nuxeo / Content Management Platform For Business Applications

Nuxeo Platform 5.8

Administration Documentation

Released on 15/04/14

Installation and Administration

Nuxeo Platform 5.8 Installation and Administration Guide

In this guide, you will find all the information to install and manage the Nuxeo Platform: how to install it, how to configure a database, install new packages from the Nuxeo Admin Center etc. The installation and administration principles described in this guide apply to all modules of the Platform: document management, digital asset management, case management, etc.

Download
Download this documentation in PDF.

For evaluation purpose

You want to evaluate or test the platform? Here the main steps you should follow to quickly install your Nuxeo application and get ready to use it.

1. Check out the requirements
2. Install
3. Start
4. Setup the platform with a preset module

Full installation – For advanced testing and production purpose

You want to install the application to use it or test it in a production environment? Follow the steps below:

- Hardware and Software Requirements
- Installation
- Setup
- Server Start and Stop
- Monitoring and Maintenance
- Nuxeo Shell
- Admin Center overview
- Marketplace Add-Ons

Hardware and Software Requirements

This section presents information about the running environment for a Nuxeo server. Listing all required software, giving a recommended configuration and listing some others, known as operational, this sections aims at helping you to validate or define your production environment. However the list is not exhaustive and needs to be completed with the users' experience.

Requirements

The Nuxeo Platform can run on Linux, Mac OS X and Windows operating systems.

All you need for a quick start is a **Java Development Kit (JDK)*** (a JRE is enough to run the Nuxeo Platform but not to develop on it). **Java 7** (also called Java 1.7) is required.

* We currently support Oracle's JDK and OpenJDK. Don't hesitate to contact us if you need us to support a JDK from another vendor.

Checking Your Java Version

To check that you have the right version of Java:

1. Open a terminal:
   - on Linux or Mac OS X: open a terminal.
   - on Windows: press "windows" key + r, type `cmd` (or `command`) in the Run window and press "OK" or open the "Prompt command" from "Start > Programs > Accessories" menu.
2. Type `java -version` and press **Enter**.
   If Java is correctly installed on your computer, the name and version of your Java virtual machine is displayed:

java -version result on Mac OS

```
java version "1.7.0_17"
Java(TM) SE Runtime Environment (build 1.7.0_17-b02)
Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01,
mixed mode)
```

java -version result on Ubuntu

```
java version "1.7.0_15"
OpenJDK Runtime Environment (IcedTea7 2.3.7)
(7u15-2.3.7-0ubuntu1~12.10.1)
OpenJDK 64-Bit Server VM (build 23.7-b01, mixed
mode)
```

If Java is not installed on you computer, it fails to display the Java version. Then, you need to install Java (see below).

If Java is installed but not included in the PATH, it fails to find the Java command. Then, you need to add `$JAVA_HOME/bin/` in your PATH (see How do I set or change the PATH system variable?).

On this page

Installing Java

For Windows Users

If the required version of Java is not installed on your computer:

1. Download it from the Oracle website and choose the appropriate platform for your hardware and Windows version.
2. Run the downloaded .exe file and follow the instructions displayed.

For Linux Users

For Ubuntu, the openjdk-7-jdk package is available from version 12.04 LTS on.

For Debian, it is not available in the current stable release (6.0.x aka "squeeze"), but it will be in the next one ("wheezy"). You can still download it from the Oracle website, but in that case you can not use the nuxeo .deb package (the zip distribution still works fine).

For Mac OS X Users

Java 7 requires at least Lion (Mac OS 10.7.3). The Java 7 package for Mac OS is available on the Oracle website and instructions for installation on the Java website.

Recommendations

Hardware Configuration

The Nuxeo Platform is designed to be scalable and can thus to be deployed on many servers. It can be installed on only one server for a start, and can also easily be installed on many servers. The constant is that there is the need to have one modern server with good performances. Then the other servers can be more lower-end.

The numbers below are given for the one needed high-end server.

- RAM: 2 GB is the minimum requirement for using Nuxeo,
- CPU: Intel Core 2 or equivalent and better.
  You might want to avoid machines from the Intel Pentium 4 Xeon series since some models have a too small amount of cache. This impairs performance greatly compared to other CPU architecture of the same generation. (Intel Pentium 4 servers are quite widespread because of an attractive price policy.)
- Storage (disk) space: the minimum Nuxeo installation, along with the needed server and libs, takes something between 200 MB and 280 MB on a filesystem. Then, the final size will of course depend on the amount of data that will be stored in Nuxeo. A safe bet (until we provide better numbers) is to consider data space ratio of 1.5 to 2.

Default Configuration

The default persistence configuration is lightweight and easy to use, but it is not made for performance.

The Nuxeo Platform uses:

- H2 for SQL Data (directories, relations ...),
- Filesystem persistence with VCS for the Document repository.

For Optimal Performances

- Linux 64 bits,
- PostgreSQL 9.0 or 9.1, Use PostgreSQL for document repository and all other services.
- Have plenty of RAM (>= 4 GB).

Known Working Configurations

OS

- Debian GNU/Linux 5.0 Lenny or more recent
- Linux Ubuntu 32 and 64 bits: 10.10 or more recent
- Linux Mandriva 2008.1
- Red Hat Linux RHEL 5 and 6
- CentOS 5
- OpenSUSE
- Mac OS X Lion (10.7), Mac OS X Mountain Lion (10.8)
- Microsoft Windows XP, Windows 7

Other Unix variants (such as Solaris) should work as long as there is an implementation of Java 7, but some adjustments may be needed to have :

- Process management running 100% ok (monitor, restart ...)
- All external converters available (OpenOffice/LibreOffice, pdf2html, ffmpeg ...)

JVM

Sun JDK 7, 64 bits recommended especially on Windows environment.

Storage Backends

Different backends may be set as well for Nuxeo Core repository as for all other Nuxeo services that persist data. Please see the list of supported databases for each version of Nuxeo.

LDAP

- OpenLDAP
- OpenDS
- Microsoft Active Directory

Browsers

Nuxeo applications can be used with the browsers below.

- IE 8 and greater with activeX enabled
- Firefox 10 and greater
- Google Chrome 8 and greater
- Safari 4 and greater

Browser extensions for Drag & Drop and Live Edit are available for Internet Explorer and Firefox only.

Installing and Setting up Related Software

The Nuxeo Platform modules use external software for some features. They need to be installed on the server in addition to Nuxeo application.

Here is the list of additional resources you may want to install:

- LibreOffice and pdftohtml: used for web preview and annotations of office documents in the Document Management module,
- ImageMagick: used for preview and tiling of picture documents in the Document Management and Digital Asset Management module,
- Ffmpeg: for Video features (needed for the Digital Asset Management module),
- Gimp and UFRaw: for RAW format images (needed for the Digital Asset Management module),
- libwpd: used for processing WordPerfect documents,
- Ghostscript: used for thumbnails generation.

Thumbnails and previews are created when documents are imported into Nuxeo, not on the fly when browsing documents. So in order to check if the third party software work properly on your Nuxeo instance, you must import new documents.

Under Debian or Ubuntu, all of this can be installed by the following command:
sudo apt-get install openjdk-7-jdk imagemagick ufraw poppler-utils libreoffice ffmpeg libwpd-tools ghostscript

Windows installer
The following software is already included when using the .exe installer:

- ffmpeg,
- ImageMagick,
- pdftohtml,
- ghostscript.

If not already present on the system, you will have the option to automatically install LibreOffice.

On this page

Setting up OpenOffice/LibreOffice and pdftohtml for Preview and Annotations on Office Documents

Installing OpenOffice/LibreOffice and pdftohtml on the server is only required if you need to use preview (and then possibly annotations) on PDF and office documents.

1. Install the following optional components:
   - PDFtoHTML (necessary for PDF documents preview and annotations), included in poppler:
     - Linux Debian or Ubuntu: sudo apt-get install poppler-utils
     - Mac OS X using HomeBrew: `brew install poppler`
     - Windows: by installing the poppler binary (available from this blogpost)
       Edit the `Path` system variable and add `;POPPLER_INSTALL_DIRECTORY` ("`C:\Program Files (x86)\Poppler`" for example).

       Old pdftohtml binaries are available from http://sourceforge.net/projects/pdftohtml/files/, but they are obsolete. It is recommended to use poppler.

- OpenOffice or LibreOffice (necessary for office documents preview and annotations):
    - Linux Debian or Ubuntu: sudo apt-get install libreoffice
    - All OS, manual download of LibreOffice 3.x or greater from http://www.libreoffice.org/ or OpenOffice 3.x or greater from http://www.openoffice.org/.

2. Start OpenOffice/LibreOffice manually once, check that it works, and install in it the additional fonts you may need for non-default languages.

> **Non-latin fonts**
> If you don't install the proper fonts for your documents (for instance Arabic or Chinese fonts), then you previews or converted documents may show questions marks or small squares instead of the characters you expect.

3. Start the OpenOffice/LibreOffice server (on a single line):

Manually start Office deamon on Windows

```
soffice.exe --headless --nofirststartwizard
--accept="socket,host=localhost,port=8100;urp;StarOffice.Service"
```

Manually start Office deamon on Linux Debian or Ubuntu

```
soffice --headless --nofirststartwizard
--accept="socket,host=localhost,port=8100;urp;StarOffice.Service" &
```

> If OpenOffice/LibreOffice is already installed on your server, Nuxeo applications come with a daemon that should automatically start it, in which case you won't need to run the above line. More information about the daemon configuration below.

> If not using default install path, you may have to add the path to the executable in your `nuxeo.conf`: `jod.office.home=/path/to/libreoffice`.

4. Restart the server after launching OpenOffice/LibreOffice server.

More information about the Nuxeo Office daemon
The deprecated OOoDaemonService has been replaced by OOoManagerService. The configuration for the new service can be found in `$NUXEO_HOME/templates/common/config/ooo-manager-config.xml`.

Setting up ImageMagick for Picture Tiling Features

The image tiling used in the preview of large images, and so for annotations, needs the installation of the ImageMagick software.

Please see Nuxeo-Book chapter about "Image tiling". Requirements (ie: ImageMagick 6.3.7 or later) are defined in the installation section.

Linux Debian or Ubuntu:

- sudo apt-get install imagemagick

Mac OS X:

- Using Homebrew: `brew install imagemagick`
- Using MacPorts: `sudo port install ImageMagick`

Windows:

- Download the ImageMagik installer from http://www.imagemagick.org/

> By default ImageMagick is multi threaded and will use all the available CPUs. This creates burst of CPU usage, especially when thumbnail is generated concurrently.
>
> Hopefully you can control the number of threads used by ImageMagick either by:
>
> - editing `/etc/ImageMagick/policy.xml` on Linux and MAC OS X, `IMAGEMAGIK_DIRECTORY/policy.xml` on Windows, and setting

```
<policy domain="resource" name="thread" value="1"/>
```

- or by adding an environment variable
  - add "`export MAGICK_THREAD_LIMIT=1`" in the nuxeo user `.bash_profile` on Linux and MAC OS X,
  - add an environment variable `MAGICK_THREAD_LIMIT` on Windows with value set to 1.

Setting up ffmpeg

To enable video features, you must install ffmpeg on the server.

Windows:

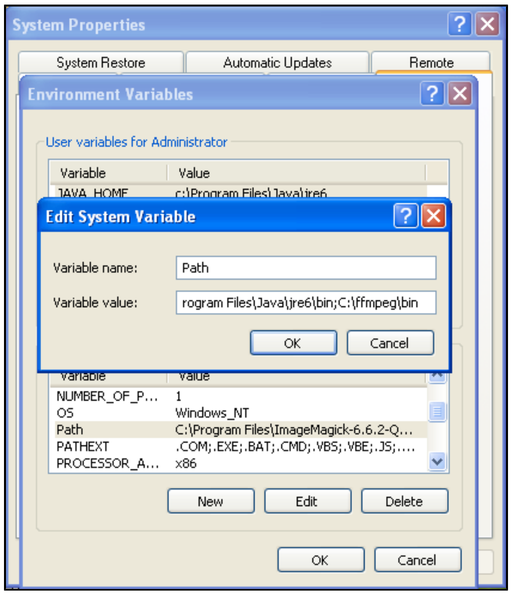
1. Download ffmpeg from [http://ffmpeg.zeranoe.com/builds/].
2. Extract the ffmpeg archive into a new folder named {{C:\ffmpeg}} for instance.

   The archives provided by this website should be decompressed with: 7-Zip

3. You have to add the ffmpeg environment variable:

- Right click on the "My Computer" icon and click on **Properties**.
- On the "Advanced" tab, edit the `Path` system variable and add `;C:\ffmpeg\bin`.

  Don't forget the semicolon at the end of existing values.



Variable	Value
JAVA_HOME	c:\Program Files\Java\jre6

Variable name: Path

Variable value: rogram Files\Java\jre6\bin;C:\ffmpeg\bin

Variable	Value
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\Program Files\ImageMagick-6.6.2-Q...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86

Linux Debian or Ubuntu:

- sudo apt-get install ffmpeg

Mac OS X:

- Using Homebrew: `brew install ffmpeg`
- Using MacPorts: `port install ffmpeg +nonfree`

Setting up Gimp and UFRaw

To enable RAW formats in a Nuxeo application, you need to download and install the following optional components:

- Gimp (needed for UFRaw) 2.6.7 or greater from Gimp Win at SourceForge
- UFRaw from UFRaw at SourceForge

Linux Debian or Ubuntu:

- sudo apt-get install ufraw

Windows

- UFRaw is available as a standalone application. You can download it from http://ufraw.sourceforge.net/Install.html#MS

Setting up libwpd

To enable processing of WordPerfect documents, you need to download and install libwpd available at SourceForge.

Linux Debian or Ubuntu:

- sudo apt-get install libwpd-tools

Setting up Ghostscript

To enable the generation of thumbnail views of documents (used in the DAM module and on the icons view), you need to install Ghostscript.

Windows:

Use the installer available from the Ghostscript download page.

Mac OS X:

- Using Homebrew: `brew install ghostscript`
- Using MacPorts: `port install ghostscript`

Linux Debian or Ubuntu:

- sudo apt-get install ghostscript

Installing Live Edit Silently

To install MSOffice Live Edit in silent mode, without any user interaction:

```
msiexec /i nuxeo-liveedit-msoffice.msi APPLICATIONFOLDER="c:/your/install/path" /qn
```

More Live Edit Admin Documentation

- Installing Live Edit Silently
- Configuration Examples
- Configuring a Reverse Proxy to Work with Live Edit and Client Certificate Authentication

Other Live Edit Related Documentation

- LiveEdit icons are still available in Nuxeo after LiveEdit has been uninstalled (Nuxeo Technical Knowledge Base (FAQ))
- LiveEdit makes MS Office slow to start (Nuxeo Technical Knowledge Base (FAQ))
- Setup Firefox protocol handler with LiveEdit 2 for MS Office and OpenOffice.org (Nuxeo Technical Knowledge Base (FAQ))
- I can't view my websites and blogs (displays a message "The HTTP header field "Accept" with value...") (Nuxeo Technical Knowledge Base (FAQ))
- Managing Your Own File with LiveEdit (Nuxeo Platform User Documentation - 5.8)
- Installing Live Edit (Nuxeo Platform User Documentation - 5.8)
- Live Edit Compatibility Table (Nuxeo Platform User Documentation - 5.8)
- Working with Live Edit (Nuxeo Platform User Documentation - 5.8)

Supported Application Servers

The Nuxeo Platform can be based on JBoss or Tomcat. Here are tables showing which versions of these application servers are known to work with the Nuxeo Platform.

JBoss

	JBoss AS 4.0.5	JBoss AS 4.2.3	JBoss EAP 5.0.0	JBoss AS 5.1.0 GA	JBoss EAP 5.1.0	JBoss EAP 5.1.1	JBoss EAP 5.1.2

Nuxeo EP 5.1.x	x						
Nuxeo EP 5.2.x		x					
Nuxeo EP 5.3.x		x					
Nuxeo EP 5.4.0				x			
Nuxeo EP 5.4.1			x	x			
Nuxeo EP 5.4.2			x	x		validation in progress	
Nuxeo Platform 5.5.x				x		validation in progress	
Nuxeo Platform 5.6				x			
Nuxeo Platform 5.8							

Apache Tomcat

	Tomcat 6.0.20	Tomcat 7.0.42
Nuxeo EP 5.1.x		
Nuxeo EP 5.2.x		
Nuxeo EP 5.3.x	x	
Nuxeo EP 5.4.x	x	
Nuxeo Platform 5.5.x	x	
Nuxeo Platform 5.6	x	
Nuxeo Platform 5.8		x (since Nuxeo 5.7.2)

Supported Databases

The Nuxeo Platform supports the following databases.

	Jackrabbit	H2	PostgreSQL	MySQL	Oracle	SQL Server	DB2
Nuxeo EP 5.1.x	1.3.3	-	-	-	-	-	-
Nuxeo EP 5.2.x	1.5.0	1.1.111	8.3 8.4	5.1	10	2005	-
Nuxeo EP 5.3.x	1.5.0	1.1.114	8.3 8.4	5.1	10	2005 2008	-
Nuxeo EP 5.4.0	-	1.1.114	8.3 8.4 9.0	5.1	10	2005 2008	-
Nuxeo EP 5.4.1	-	1.1.114	8.3 8.4 9.0	5.1	10	2005 2008	-
Nuxeo EP 5.4.2	-	1.1.114	8.3 8.4 9.0 9.1	5.1	10	2005 2008	-

Nuxeo Platform 5.5.x	-	1.1.114	8.4 9.0 9.1	5.1	10 11	2005 2008	-
Nuxeo Platform 5.6.x	-	1.1.114	8.4 9.0 9.1 9.2	5.1 5.5 5.5 (Amazon RDS)	10 11 11 (Amazon RDS)	2005 2008 2008r2 2012	-
Nuxeo Platform 5.8	-	1.1.114	8.4 9.0 9.1 9.2 9.3	5.1 5.5 5.5 (Amazon RDS)	10 11 11 (Amazon RDS)	2008 2008r2 2012 2012 (Azure)	

Note that only the latest service pack is supported for a given version. For the open source databases this means upgrading to the latest minor version (ex: MySQL 5.5.28 or PostgreSQL 9.2.1 at the time of this writing). For For Oracle this means the latest patchset (ex: 11.2.0.3 at the time of this writing). For SQL Server this means the latest service pack (ex: Service Pack 2, 10.50.4000 at the time of this writing).

Note that the exact version numbers for versions before 5.6 may be slightly off, we're in the process of reviewing past data.

Installation

The Nuxeo Platform comes in different packages and can be installed on any operating system. You may have to install:

- a zip archive (works on any operating system),
- a Windows installer (.exe),
- a virtual machine image (works on any operating system),
- a .deb package (works on Linux Debian and Ubuntu).

Our installation recipies:

- Installing the Nuxeo Platform on Mac OS
- Installing the Nuxeo Platform on Windows
    - Installing the Nuxeo Platform as a Windows Service
    - Running Multiple Server Instances in Windows
- Installing the Nuxeo Platform on Linux
    - Configuring the Nuxeo Platform as a Daemon on Debian
- Deploying Nuxeo on Amazon AWS
- Deploying Nuxeo on CloudFoundry
- How to Estimate Disk and Database Usage

Installing the Nuxeo Platform on Mac OS

On Mac OS, you can install the Nuxeo Platform using two different packages:

- the .zip archive,
- the virtual machine image.

How to Install the Nuxeo Platform From the .zip Archive

Installing the Nuxeo Platform using the .zip package installs the Nuxeo Platform only. External dependencies must be installed separately.

To install the Nuxeo Platform zip archive:
Unzip the .zip archive using your favorite tool.

What's next?
You want to evaluate the application? You can now start the server.
You want to do a complete installation, compatible for a production environment? You should now prepare your environment.

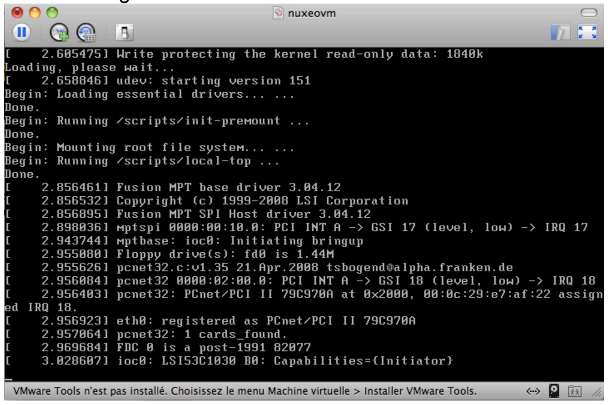
How to install a Nuxeo Virtual Machine Image

The Nuxeo Platform is available as ready-to-use virtual machine images from nuxeo.com. VM images are available for VMWare and Virtual Box. They provide a full environment (OS, database...) and all required dependencies to make the Nuxeo Platform work.

To install the Nuxeo virtual machine image and start Nuxeo:

1. Unzip the downloaded package.
   You get a folder with the required file for the virtual machine image to run.
2. Start the virtual machine image in your virtual machine application by double-clicking on it.
    - For the VMWare package, double-click on the file "nuxeo.vmx".
    - For the OVF package, double-click on the .ovf file that corresponds to the supported standard: "nuxeo_OVF10.ovf" for Open Virtualization Format 1.0, supported by Virtual Box for instance, or "nuxeo_OVF09.ovf" for Open Virtualization Format 0.9. Then start the imported virtual machine.

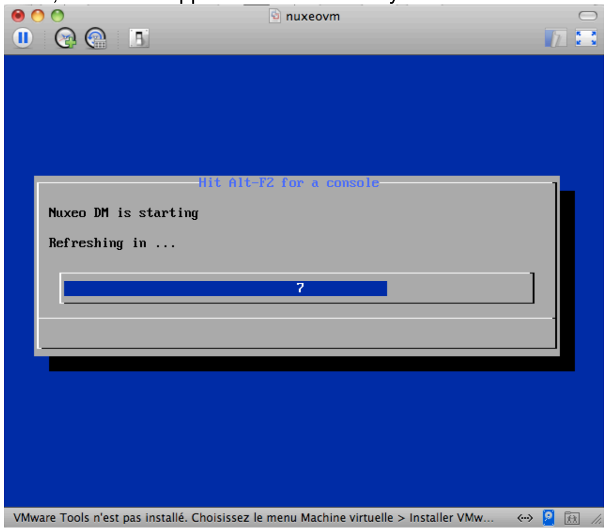
The VM image starts.



```
[    2.605475] Write protecting the kernel read-only data: 1840k
Loading, please wait...
[    2.658846] udev: starting version 151
Begin: Loading essential drivers... ...
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
[    2.856461] Fusion MPT base driver 3.04.12
[    2.856532] Copyright (c) 1999-2008 LSI Corporation
[    2.856895] Fusion MPT SPI Host driver 3.04.12
[    2.898036] mptspi 0000:00:10.0: PCI INT A -> GSI 17 (level, low) -> IRQ 17
[    2.943744] mptbase: ioc0: Initiating bringup
[    2.955080] Floppy drive(s): fd0 is 1.44M
[    2.955626] pcnet32.c:v1.35 21.Apr.2008 tsbogend@alpha.franken.de
[    2.956084] pcnet32 0000:02:00.0: PCI INT A -> GSI 18 (level, low) -> IRQ 18
[    2.956403] pcnet32: PCnet/PCI II 79C970A at 0x2000, 00:0c:29:e7:af:22 assign
ed IRQ 18.
[    2.956923] eth0: registered as PCnet/PCI II 79C970A
[    2.957064] pcnet32: 1 cards_found.
[    2.969684] FDC 0 is a post-1991 82077
[    3.028607] ioc0: LSI53C1030 B0: Capabilities={Initiator}
_
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMware Tools.

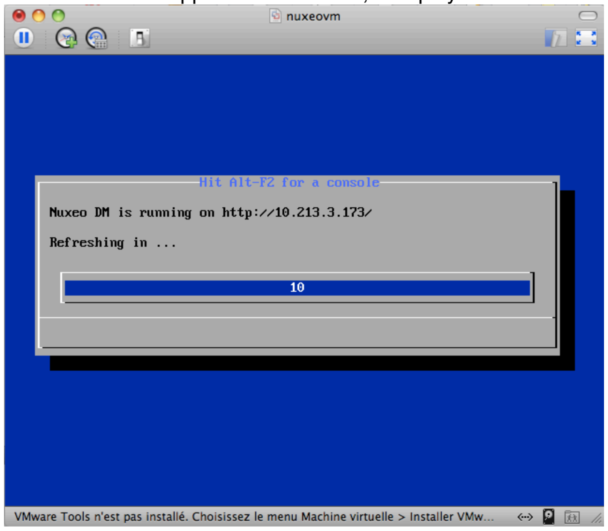
Then, the Nuxeo application automatically starts.



```
Hit Alt-F2 for a console

Nuxeo DM is starting

Refreshing in ...

7
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

When the Nuxeo application is started, it displays the address at which it is available.



```
Hit Alt-F2 for a console

Nuxeo DM is running on http://10.213.3.173/

Refreshing in ...

10
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

3. In your browser, type the indicated address.
   The startup wizard is displayed to help you configure your application.

> **Shell root access**
> The password for the root and nuxeo users are generated the first time you start the virtual machine and are displayed on the console.

Installing the Nuxeo Platform on Windows

You can install the Nuxeo Platform on Windows using one of several methods:

- the Windows installer (.exe),

- the .zip archive,
- the virtual machine image.

How to Install the Nuxeo Platform from the Windows Installer

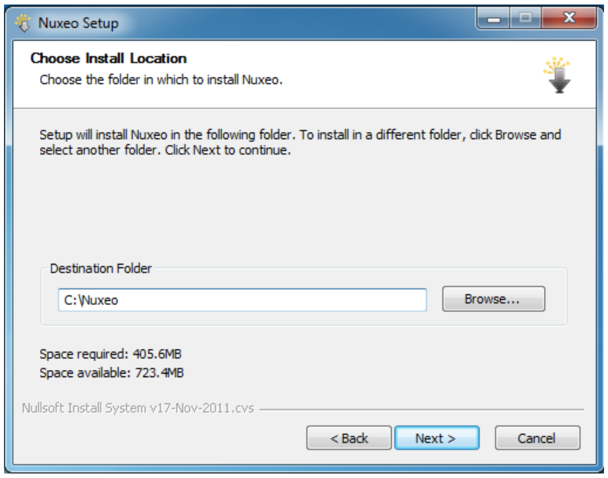
The Nuxeo Platform is available with a Windows installer that guides you in the install process.

The Windows installer requires Administrator privileges.

On this page

To install the application using the Windows installer (.exe), double-click on the .exe installer you downloaded and follow the instructions displayed.

On Windows in general, and especially on Windows 7, it is highly recommended to install your Nuxeo application at the root of a disk (`C:\Nuxeo` for instance), because of rights issues, limitations on paths length, 32/64 bits conflicts,... An installation in another folder could provoke restart issues at end of the startup wizard steps.


Nuxeo Setup

Choose Install Location
Choose the folder in which to install Nuxeo.

Setup will install Nuxeo in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder
C:\Nuxeo

Space required: 405.6MB
Space available: 723.4MB

Nullsoft Install System v17-Nov-2011.cvs

What's next?
You want to evaluate the application? You can now start the server.
You want to do a complete installation, compatible for a production environment? You should now prepare your environment.

The Windows installer includes a full JDK installation. You will get an error while installing the **JavaFX** portion. The JavaFX platform is not used by Nuxeo, therefore, when the error appears, click **Close** and continue with the installation.

When you launch the Nuxeo services for the first time, you will see an error window that says **pdftohtml.exe has stopped working**. To get the pdftohtml utility to work properly, install the **Visual C++ Redistributable for Visual Studio 2012** available on Microsoft's website.

If you are using PostgreSQL, please review the Configuring PostgreSQL document and use the recommended configurations.

How to Install the Nuxeo Platform from the .zip Archive

Installing the Nuxeo Platform using the .zip package installs the Nuxeo Platform only. External dependencies must be installed separately.

To install the Nuxeo Platform zip archive:

Unzip the .zip archive using your favorite tool.

On Windows in general, and especially on Windows 7, it is highly recommended to install your Nuxeo application at the root of a disk (`C:\Nuxeo` for instance), because of rights issues, limitations on paths length, 32/64 bits conflicts,... An installation in another folder could provoke restart issues at end of the startup wizard steps.

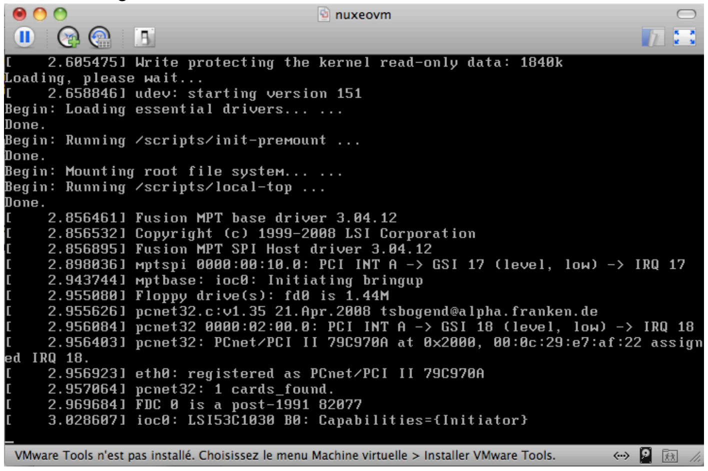
What's next?
You want to evaluate the application? You can now start the server.
You want to do a complete installation, compatible for a production environment? You should now prepare your environment.

How to Install a Nuxeo Virtual Machine Image

The Nuxeo Platform is available as ready-to-use virtual machine images from nuxeo.com. VM images are available for VMWare and Virtual Box. They provide a full environment (OS, database…) and all required dependencies to make the Nuxeo Platform work.

To install the Nuxeo virtual machine image and start Nuxeo:

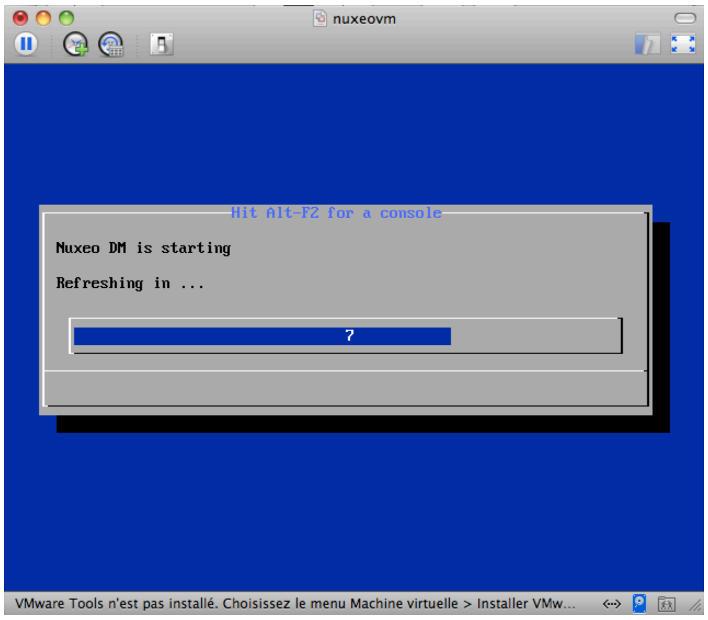
1. Unzip the downloaded package.
   You get a folder with the required file for the virtual machine image to run.
2. Start the virtual machine image in your virtual machine application by double-clicking on it.
   - For the VMWare package, double-click on the file "nuxeo.vmx".
   - For the OVF package, double-click on the .ovf file that corresponds to the supported standard: "nuxeo_OVF10.ovf" for Open Virtualization Format 1.0, supported by Virtual Box for instance, or "nuxeo_OVF09.ovf" for Open Virtualization Format 0.9. Then start the imported virtual machine.

   The VM image starts.
   

```
[    2.605475] Write protecting the kernel read-only data: 1840k
Loading, please wait...
[    2.658846] udev: starting version 151
Begin: Loading essential drivers... ...
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
[    2.856461] Fusion MPT base driver 3.04.12
[    2.856532] Copyright (c) 1999-2008 LSI Corporation
[    2.856895] Fusion MPT SPI Host driver 3.04.12
[    2.898036] mptspi 0000:00:10.0: PCI INT A -> GSI 17 (level, low) -> IRQ 17
[    2.943744] mptbase: ioc0: Initiating bringup
[    2.955080] Floppy drive(s): fd0 is 1.44M
[    2.955626] pcnet32.c:v1.35 21.Apr.2008 tsbogend@alpha.franken.de
[    2.956084] pcnet32 0000:02:00.0: PCI INT A -> GSI 18 (level, low) -> IRQ 18
[    2.956403] pcnet32: PCnet/PCI II 79C970A at 0x2000, 00:0c:29:e7:af:22 assign
ed IRQ 18.
[    2.956923] eth0: registered as PCnet/PCI II 79C970A
[    2.957064] pcnet32: 1 cards_found.
[    2.969684] FDC 0 is a post-1991 82077
[    3.028607] ioc0: LSI53C1030 B0: Capabilities={Initiator}
_
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMware Tools.

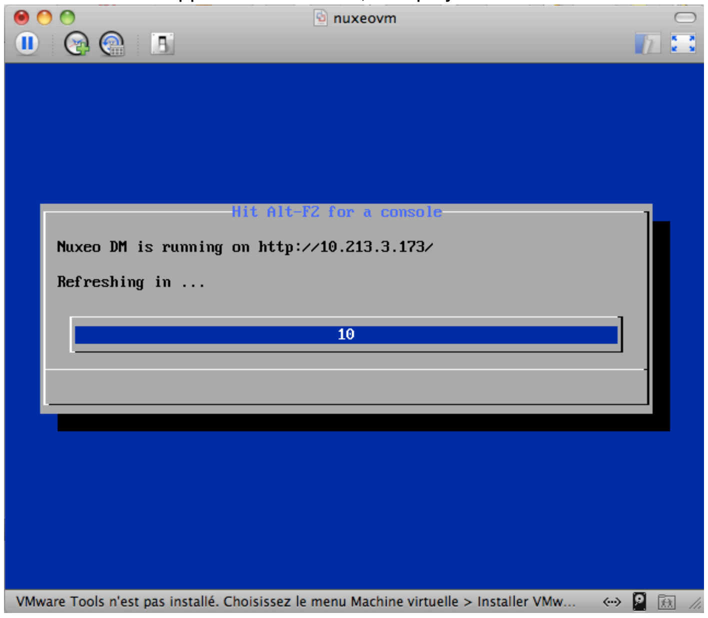
   Then, the Nuxeo application automatically starts.



```
Hit Alt-F2 for a console

Nuxeo DM is starting

Refreshing in ...

7
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

When the Nuxeo application is started, it displays the address at which it is available.



```
Hit Alt-F2 for a console

Nuxeo DM is running on http://10.213.3.173/

Refreshing in ...

10
```

VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

3. In your browser, type the indicated address.
   The startup wizard is displayed to help you configure your application.

> **Shell root access**
> The password for the root and nuxeo users are generated the first time you start the virtual machine and are displayed on the console.

Related pages:

- Installing the Nuxeo Platform on Windows
- Running Multiple Server Instances in Windows
- Installing the Nuxeo Platform as a Windows Service

Installing the Nuxeo Platform as a Windows Service

Installing Nuxeo as a Windows service is independent of Nuxeo. So, this is no longer in our development scope since Nuxeo 5.4.

Multiple solutions are available, here are some of them, given without any warranty.

Once a batch is installed as a service, it cannot be changed: you must first uninstall it, then edit and reinstall in order to change its content.
So, it's generally a good idea to write a batch file wrapping calls to `nuxeoctl.bat` and install that `NuxeoWrapper.bat` as a service, which will be responsible of starting Nuxeo with the wanted user and environment parameters.

Prerequisites

In order to run as a service, you have to manage the directory rights for the super-user running the service. There are behavior changes depending on the Windows version.

Also, take care that network directories are usually not available when a service is executing. So, if you need to use some, you will have to mount them in the batch script before starting Nuxeo.

The database used by Nuxeo has to be installed as a service and started before the Nuxeo service.

On this page

Available Solutions

Yet Another Java Service Wrapper (Recommended)

YAJSW is a Java centric implementation of the Java Service Wrapper by tanuki (JSW).
It aims at being mostly configuration compliant with the original. YAJSW is LGPL licensed.
That solution seems to be the more flexible, robust and multi-OS compliant.

When using this solution it is important that Nuxeo installation path contains no space (typically NOT C:\Program Files\nuxeo). Otherwise the service wrapper will truncate the path after the space and the service will not start. It is recommended that Nuxeo is installed at the root of the volume (C:\nuxeo\ for instance).

Installing Nuxeo as a Windows Service Using YAJSW

1. Download YAJSW and unzip the archive.
2. Set the system environment variable `NUXEO_CONF` to the location of your `nuxeo.conf` file, something like `%NUXEO_HOME%\bin\nuxeo.conf`.
3. Start Nuxeo DM from the command line:

```
nuxeoctl.bat --gui=false start
```

   Once the server is started, you'll get a message like below where XXXX is the process ID of the running Nuxeo application:

```
Server started with process ID XXXX.
```

4. Start a Command Prompt as an Administrator.
5. Go to the `%YAJSW_HOME%\bat` folder.
6. Execute the `genConfig`command with the process ID as parameter:

```
genConfig.bat XXXX
```

   The configuration is written in the file `%YAJSW_HOME%\conf\wrapper.conf`.

Wrapper Static Configuration Vs nuxeo.conf
Once the wrapper configuration is generated, it is static. It means any further start of the Nuxeo service won't read nuxeo.conf t o apply configuration changes to `*config.xml` files or to JVM options.

If you change nuxeo.conf after having installing Nuxeo as a service, there are two possibilities:

- your changes are only about the configuration of Nuxeo: you just need to start Nuxeo with its own launcher (step 3);
- your changes are about JVM options (changes in `JAVA_OPTS` variable: memory, debug ...): you'll have to start Nuxeo with its launcher and generate the new wrapper configuration (step 3 to 6).

In both cases, you don't have to unregister the service and register it again: the wrapper will use the same configuration file (wrapper.conf).

7. Stop Nuxeo DM:

```
nuxeoctl.bat --gui=false stop
```

8. Open the wrapper.conf file and make sure the `wrapper.console.title` entry does not contain a colon (":") or the runConsole.bat will fail. Just remove the colon or give another title.
9. Execute your wrapped application as console application by calling this command and check your application is accessible:

```
runConsole.bat
```

10. Edit the file `%YAJSW_HOME%\conf\wrapper.conf`and set your custom values for these parameters:

```
# Name of the service
wrapper.ntservice.name=NuxeoDM
# Display name of the service
wrapper.ntservice.displayname=Nuxeo DM
# Description of the service
wrapper.ntservice.description=Service to manage Nuxeo DM
```

11. To install the application as service call, execute:

```
installService.bat
```

Your service is installed and you can run Nuxeo DM from its service ("Windows Computer Management > Services" on Windows 7).

JBoss Native Windows (aka JBossSVC, JBossService and JavaService)

Deprecated Nuxeo scripts managing install as a Windows service were previously used. They were based on JBoss Native Windows which is now not recommended because of a number of defects. However, it was relatively easy to use and provides a quick solution.

As an example, here is the content of jboss-native-2.0.4/bin/service.bat:

Nuxeo JBoss Service Script for Windows

```
@echo off
REM JBoss, the OpenSource webOS
REM
REM Distributable under LGPL license.
REM See terms of license at gnu.org.
REM
REM -------------------------------------------------------------------------
REM JBoss Service Script for Windows
REM -------------------------------------------------------------------------
```

```
@if not "%ECHO%" == "" echo %ECHO%
@if "%OS%" == "Windows_NT" setlocal
set DIRNAME=%CD%

REM
REM VERSION, VERSION_MAJOR and VERSION_MINOR are populated
REM during the build with ant filter.
REM
set SVCNAME=NuxeoEP
set SVCDISP=NuxeoEP
set SVCDESC=Nuxeo 5.3.0-GA / JBoss Application Server 4.2.3 GA / Platform: Windows 64
set NOPAUSE=Y

REM Suppress killing service on logoff event
set JAVA_OPTS=-Xrs

REM Figure out the running mode

if /I "%1" == "install"   goto cmdInstall
if /I "%1" == "uninstall" goto cmdUninstall
if /I "%1" == "start"     goto cmdStart
if /I "%1" == "stop"      goto cmdStop
if /I "%1" == "restart"   goto cmdRestart
if /I "%1" == "signal"    goto cmdSignal
echo Usage: service install^|uninstall^|start^|stop^|restart^|signal
goto cmdEnd

REM jbosssvc retun values
REM ERR_RET_USAGE           1
REM ERR_RET_VERSION         2
REM ERR_RET_INSTALL         3
REM ERR_RET_REMOVE          4
REM ERR_RET_PARAMS          5
REM ERR_RET_MODE            6

:errExplain
if errorlevel 1 echo Invalid command line parameters
if errorlevel 2 echo Failed installing %SVCDISP%
if errorlevel 4 echo Failed removing %SVCDISP%
if errorlevel 6 echo Unknown service mode for %SVCDISP%
goto cmdEnd

:cmdInstall
jbosssvc.exe -imwdc %SVCNAME% "%DIRNAME%" "%SVCDISP%" "%SVCDESC%" service.bat
if not errorlevel 0 goto errExplain
echo Service %SVCDISP% installed
goto cmdEnd

:cmdUninstall
jbosssvc.exe -u %SVCNAME%
if not errorlevel 0 goto errExplain
echo Service %SVCDISP% removed
goto cmdEnd

:cmdStart
REM Executed on service start
del .r.lock 2>&1 | findstr /C:"being used" > nul
```

```
if not errorlevel 1 (
  echo Could not continue. Locking file already in use.
  goto cmdEnd
)
echo Y > .r.lock
jbosssvc.exe -p 1 "Starting %SVCDISP%" > run.log
call run.bat -b 0.0.0.0 < .r.lock >> run.log 2>&1
jbosssvc.exe -p 1 "Shutdown %SVCDISP% service" >> run.log
del .r.lock
goto cmdEnd

:cmdStop
REM Executed on service stop
echo Y > .s.lock
jbosssvc.exe -p 1 "Shutting down %SVCDISP%" > shutdown.log
call shutdown -S < .s.lock >> shutdown.log 2>&1
jbosssvc.exe -p 1 "Shutdown %SVCDISP% service" >> shutdown.log
del .s.lock
goto cmdEnd

:cmdRestart
REM Executed manually from command line
REM Note: We can only stop and start
echo Y > .s.lock
jbosssvc.exe -p 1 "Shutting down %SVCDISP%" >> shutdown.log
call shutdown -S < .s.lock >> shutdown.log 2>&1
del .s.lock
:waitRun
REM Delete lock file
del .r.lock > nul 2>&1
REM Wait one second if lock file exist
jbosssvc.exe -s 1
if exist ".r.lock" goto waitRun
echo Y > .r.lock
jbosssvc.exe -p 1 "Restarting %SVCDISP%" >> run.log
call run.bat < .r.lock >> run.log 2>&1
jbosssvc.exe -p 1 "Shutdown %SVCDISP% service" >> run.log
del .r.lock
goto cmdEnd

:cmdSignal
REM Send signal to the service.
REM Requires jbosssch.dll to be loaded in JVM
@if not ""%2"" == """" goto execSignal
echo Missing signal parameter.
echo Usage: service signal [0...9]
goto cmdEnd
:execSignal
jbosssvc.exe -k%2 %SVCNAME%
goto cmdEnd
```

```
:cmdEnd
```

Other implementations were available from JBoss.

⚠ They were licensed under LGPL and so redistributable but there are not fully satisfying.

Tomcat Service Install/Uninstall Script

Using the Tomcat distribution of Nuxeo, you will find a `service.bat` script in the `bin` directory that could be adapted to install Nuxeo as a Windows service.

JavaServiceWrapper by Tanuki

Tanuki's library provides multiple methods for integrating a software as a service on various OS, the easier is to use the WrapperSimpleApp helper class to launch the application: see the example of JBoss installed as a Windows service.
It requires to unzip the downloaded wrapper file, configure a `wrapper.conf` file pointing to `%NUXEO_HOME%\bin\nuxeoctl.bat`, then write a `wrapper.bat` file for managing test/install/uninstall:

JavaServiceWrapper usage

```
REM Test:
wrapper.exe -c %NUXEO_HOME%\wrapper\wrapper.conf

REM Install:
wrapper.exe -i %NUXEO_HOME%\wrapper\wrapper.conf

REM Uninstall:
wrapper.exe -r %NUXEO_HOME%\wrapper\wrapper.conf
```

This solution is known to work well but is sadly not redistributable for us because of its GPL/Commercial license.

.NET InstallUtil

.NET framework provides an `InstallUtil.exe` tool for installing/uninstalling services.

InstallUtil usage

```
REM Install
InstallUtil /i %NUXEO_HOME\bin\service.bat

REM Uninstall
InstallUtil /u %NUXEO_HOME\bin\service.bat
```

⚠ There are some disadvantages such as failures in case of multiple frameworks installed and frontward/backward incompatibilities.

You may want to have a look at http://msdn2.microsoft.com/en-US/library/system.configuration.install.managedinstallerclass.aspx for managing that programmatically.

Related pages

- Installing the Nuxeo Platform as a Windows Service
- Installing the Nuxeo Platform on Windows
- Running Multiple Server Instances in Windows

Running Multiple Server Instances in Windows

The location of the `nuxeo.conf` is defined by that order of priority (i.e. first one of those found is used):

- Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\PRODNAME\ConfigFile` with PRODNAME equals "Nuxeo CAP", "Nuxeo DM", "Nuxeo DAM", ...
- Environment variable `NUXEO_CONF`
- `"nuxeo.conf"` file in the working directory
- `"nuxeo.conf"` file on the Desktop
- `"nuxeo.conf"` file in the same location as the (real) NuxeoCtl.exe (for versions<5.4.1) or nuxeoctl.bat (for versions5.4.1).

To launch multiple instances of Nuxeo you'd need to remove the registry key (set up by the Windows installer) and have wrappers around `NuxeoCtl.exe/nuxeoctl.bat` that define different `NUXEO_CONF` environment variables.

Note that you'd also want to have different `nuxeo.data.dir`, `nuxeo.log.dir`, `nuxeo.tmp.dir`, `nuxeo.server.http.port` and `nuxeo.server.tomcat-admin.port` in the two `nuxeo.conf` files (you can set `nuxeo.server.ajp.port` to 0 to disable AJP if you don't use it).

Related Installation on Windows Documentation

Installing the Nuxeo Platform on Windows

Running Multiple Server Instances in Windows

Installing the Nuxeo Platform as a Windows Service

Installing the Nuxeo Platform on Linux

On Linux, you can install the Nuxeo Platform using the packages below:

- the .zip archive,
- the virtual machine image,
- from the APT repository for Debian and Ubuntu.

How to Install the Nuxeo Platform from the .zip Archive

Installing the Nuxeo Platform using the .zip package installs the Nuxeo Platform only. External dependencies must be installed separately.

To install the Nuxeo Platform zip archive:
Unzip the .zip archive using your favorite tool.

What's next?
You want to evaluate the platform? You can now start the server.
You want to do a complete installation, compatible for a production environment? You should now prepare your environment.

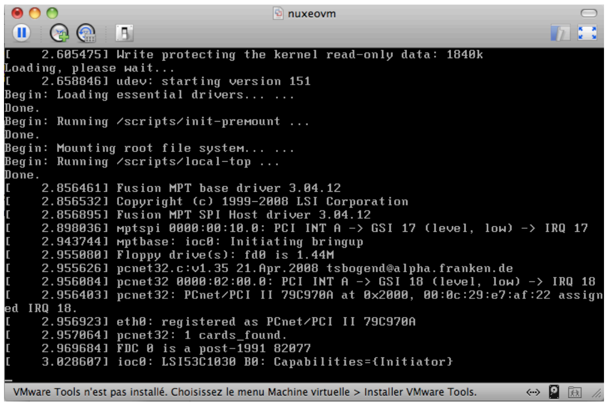
On this page

How to Install a Nuxeo Virtual Machine Image

The Nuxeo Platform is available as ready-to-use virtual machine images from nuxeo.com. VM images are available for VMWare and Virtual Box. They provide a full environment (OS, database...) and all required dependencies to make the Nuxeo Platform work.

To install the Nuxeo virtual machine image and start Nuxeo:

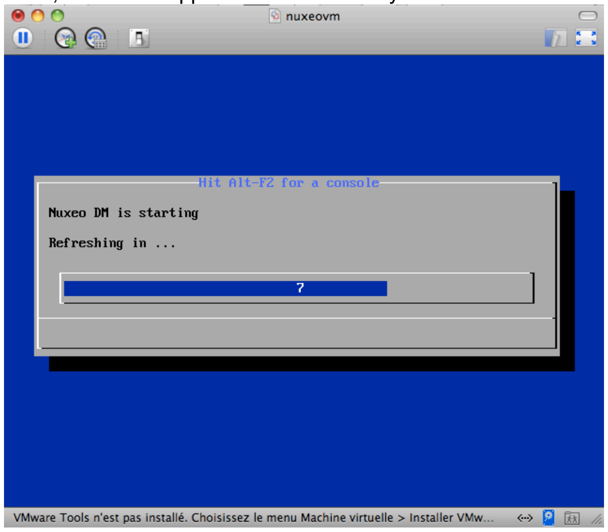
1. Unzip the downloaded package.
   You get a folder with the required file for the virtual machine image to run.
2. Start the virtual machine image in your virtual machine application by double-clicking on it.
    - For the VMWare package, double-click on the file "nuxeo.vmx".
    - For the OVF package, double-click on the .ovf file that corresponds to the supported standard: "nuxeo_OVF10.ovf" for Open Virtualization Format 1.0, supported by Virtual Box for instance, or "nuxeo_OVF09.ovf" for Open Virtualization Format 0.9. Then start the imported virtual machine.
      The VM image starts.



```
[    2.605475] Write protecting the kernel read-only data: 1840k
Loading, please wait...
[    2.658846] udev: starting version 151
Begin: Loading essential drivers... ...
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
[    2.856461] Fusion MPT base driver 3.04.12
[    2.856532] Copyright (c) 1999-2008 LSI Corporation
[    2.856895] Fusion MPT SPI Host driver 3.04.12
[    2.898036] mptspi 0000:00:10.0: PCI INT A -> GSI 17 (level, low) -> IRQ 17
[    2.943744] mptbase: ioc0: Initiating bringup
[    2.955080] Floppy drive(s): fd0 is 1.44M
[    2.955626] pcnet32.c:v1.35 21.Apr.2008 tsbogend@alpha.franken.de
[    2.956084] pcnet32 0000:02:00.0: PCI INT A -> GSI 18 (level, low) -> IRQ 18
[    2.956403] pcnet32: PCnet/PCI II 79C970A at 0x2000, 00:0c:29:e7:af:22 assign
ed IRQ 18.
[    2.956923] eth0: registered as PCnet/PCI II 79C970A
[    2.957064] pcnet32: 1 cards_found.
[    2.969684] FDC 0 is a post-1991 82077
[    3.028607] ioc0: LSI53C1030 B0: Capabilities={Initiator}
```

nuxeovm
VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMware Tools.

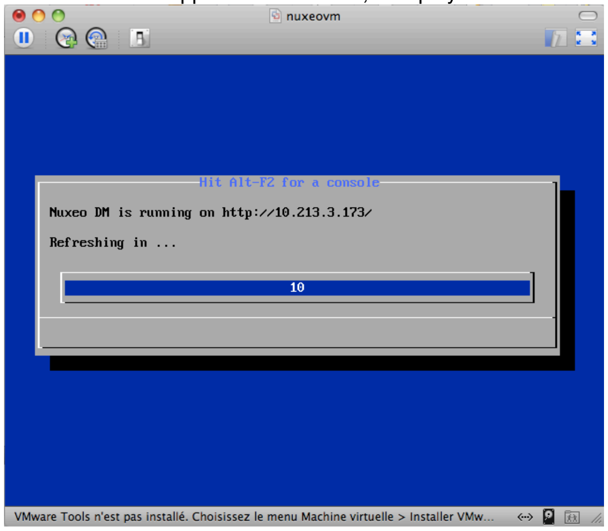
Then, the Nuxeo application automatically starts.



```
Hit Alt-F2 for a console

Nuxeo DM is starting

Refreshing in ...

7
```

nuxeovm
VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

When the Nuxeo application is started, it displays the address at which it is available.



```
Hit Alt-F2 for a console

Nuxeo DM is running on http://10.213.3.173/

Refreshing in ...

10
```

nuxeovm
VMware Tools n'est pas installé. Choisissez le menu Machine virtuelle > Installer VMw...

3. In your browser, type the indicated address.
   The startup wizard is displayed to help you configure your application.

> **Shell root access**
> The password for the root and nuxeo users are generated the first time you start the virtual machine and are displayed on the console.

How to Install the Nuxeo Platform from the APT Repository for Debian and Ubuntu

Installing the Nuxeo Platform using the APT sources for Debian and Ubuntu installs and configures the platform, but it also installs all required dependencies for an optimal use of the platform.

You can either install the Nuxeo Platform using the OS graphical user interface or from the terminal.

You will need to know two things first:

- the codename of your distribution (eg **precise** for Ubuntu 12.04 LTS)
- which kind of Nuxeo release you want to install (LTS or Fast Track, see this page for more details about the Nuxeo release cycle).

For the examples below, let's say you are using Ubuntu 12.04 LTS ("precise") and want to use the Nuxeo latest LTS release ("releases" repository, for Fast Track releases replace with "fasttracks").

Installing from the APT Sources Using the User Graphical Interface

This requires X11.

1. Edit the "Software sources" (run "`gksudo software-properties-gtk`",use the Unity Dash or browse "`System/Administration/Software Sources`" Gnome 2 menu).
2. Download the Nuxeo key and import it in the "Authentication" tab.
3. Add the Nuxeo APT repository: on the "Other Software" tab, add "`deb http://apt.nuxeo.org/ precise releases`" to the sources. (if you're using another version of Ubuntu, replace precise by the adequate name, for instance **raring** for Ubuntu 13.04)
4. Click on: apt://nuxeo .
5. Follow the instructions displayed.
   If it's your first install, you can configure:
   - the bind address,
   - the port,
   - the database (a preconfigured PostgreSQL database is suggested by default).
     The platform is installed as a service. It is automatically started and set to automatically start at boot.
6. Open a browser and type the URL http://localhost:8080/nuxeo/.
   The startup wizard is displayed so you can setup your Nuxeo platform and select the module you want to install.

Installing from the APT Sources from the Terminal

1. Import the Nuxeo key.

```
wget -q -O- http://apt.nuxeo.org/nuxeo.key | sudo apt-key add -
```

2. Add the Nuxeo APT repository.

```
sudo add-apt-repository "deb http://apt.nuxeo.org/ precise fasttracks"
```

   If you don't have `add-apt-repository`, which is a non-standard command, create a file named "`/etc/apt/sources.list.d/nuxeo.list`" and write into it:

```
deb http://apt.nuxeo.org/ precise fasttracks
```

3. Update your APT cache.

```
sudo apt-get update
```

4. Install the Nuxeo Platform.

```
sudo apt-get install nuxeo
```

5. Follow the instructions displayed. If it's your first install, you can configure:
   - the bind address,
   - the port,

- the database (a preconfigured PostgreSQL database is suggested by default).

The platform is installed as a service. It is automatically started and set to automatically start at boot.

6. Open a browser and type the URL http://localhost:8080/nuxeo/. The startup wizard is displayed so you can setup your Nuxeo platform and select the module you want to install.

See Configuring Nuxeo Debian or Ubuntu repositories for more explanations on those command lines.

Related pages

- Installing the Nuxeo Platform on Linux
- Configuring the Nuxeo Platform as a Daemon on Debian

Configuring the Nuxeo Platform as a Daemon on Debian

The procedure described here is targeted for the Debian Etch distribution, and should be valid for any Debian-based GNU/Linux distribution such as Ubuntu. In other GNU/Linux distributions some commands may be different.

Here is a sample script based on the one used in the debian package

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          nuxeo
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start/stop Nuxeo
### END INIT INFO

DESC="Nuxeo"

NUXEO_USER=nuxeo
NUXEOCTL="/var/lib/nuxeo/server/bin/nuxeoctl"
NUXEO_CONF="/etc/nuxeo/nuxeo.conf"
export NUXEO_CONF

. /lib/init/vars.sh
. /lib/lsb/init-functions


create_pid_dir() {
    mkdir -p /var/run/nuxeo
    chown $NUXEO_USER:$NUXEO_USER /var/run/nuxeo
}

# Change ulimit to minimum needed by Nuxeo
ulimit -n 2048

case "$1" in
  start)
        log_daemon_msg "Starting" "$DESC\n"
        create_pid_dir
        su $NUXEO_USER -c "$NUXEOCTL --quiet startbg"
        ES=$?
        log_end_msg $ES
        ;;
  stop)
        log_daemon_msg "Stopping" "$DESC\n"
        su $NUXEO_USER -c "$NUXEOCTL --quiet stop"
        ES=$?
        log_end_msg $ES
        ;;
restart)
        create_pid_dir
        su $NUXEO_USER -c "$NUXEOCTL --quiet restart"
        ES=$?
        log_end_msg $ES
        ;;
force-reload)
        create_pid_dir
        su $NUXEO_USER -c "$NUXEOCTL --quiet restart"
        ES=$?
        log_end_msg $ES
        ;;
status)
        su $NUXEO_USER -c "$NUXEOCTL --quiet status"
        exit $?
        ;;
*)
        echo "Usage: $0 {start|stop|restart|force-reload|status}" >&2
        exit 3
        ;;
esac
```

Copy the shell script to `/etc/init.d/nuxeo`, replacing paths to match your installation.

Then enable the autostart creating the links in the rcX.d directories running the command (as root):

```
$ update-rc.d nuxeo defaults
```

Now restart the machine and verify that nuxeo is started automatically looking at the log file.

If you want to remove the automatic startup use the command (as root):

```
$ update-rc.d -f nuxeo remove
```

You can manage the service with the following command:

```
/etc/init.d/nuxeo [status|start|stop|...]
```

Deploying Nuxeo on Amazon AWS

Need a quick Nuxeo instance for your cloud? You can deploy one in just a few minutes with our CloudFormation template, as we provide a template that will automatically install the latest Nuxeo on your Amazon AWS and all the required resources.

Prerequisites

You need an account on Amazon AWS with the CloudFormation service activated.
To sign up for AWS, just go to http://aws.amazon.com/ and click on the “Sign Up Now” link.

To activate the CloudFormation service, sign in to your management console, click on the “CloudFormation” tab and follow the instructions.

If you don't have a keypair, you will also want to create a one so you can connect to your instance later. You can create one in the “EC2” tab in your management console.

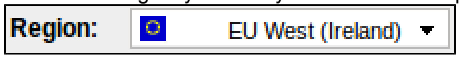
You're ready to deploy our template!

Deploying the Template

Deploying the Nuxeo template on Amazon AWS installs:

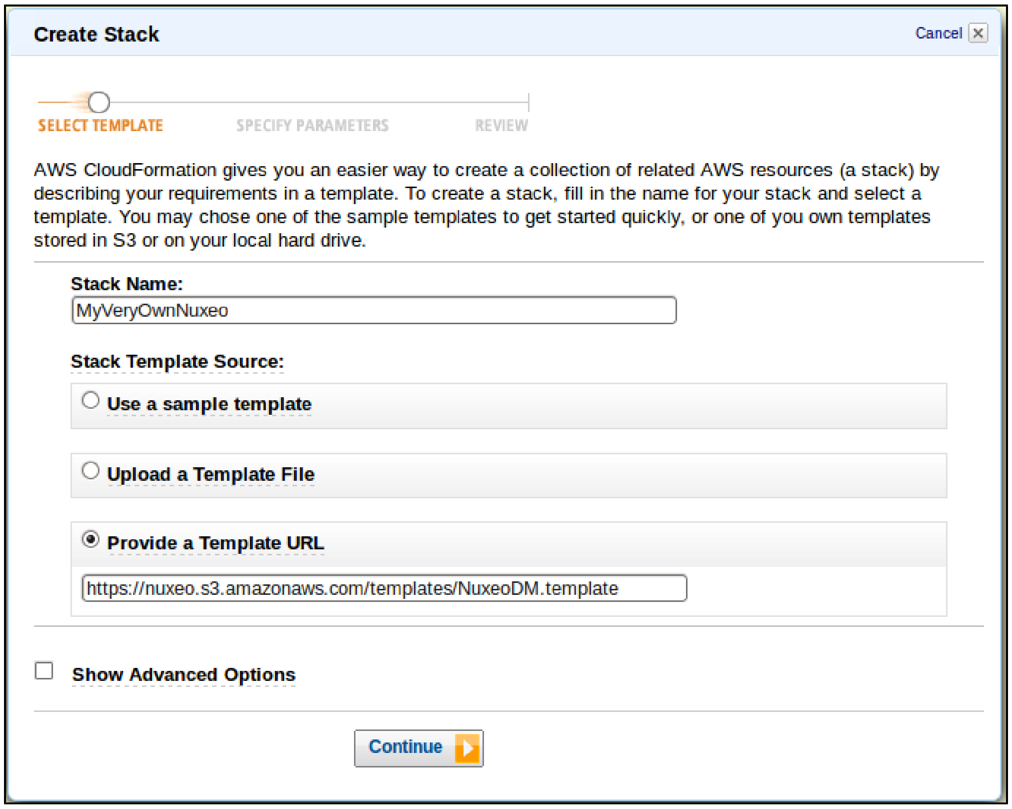
- The latest version of Nuxeo, with a PostgreSQL database and an Apache2 HTTP front-end;
- All the required Amazon resources, which are: an EC2 instance, an elastic IP, an EBS volume.

To deploy the Nuxeo template:

1. Sign in to your CloudFormation management console.
2. Choose the region you want your stack to be deployed in.


3. Start the new stack creation by clicking the "Create New Stack" button.

4. Choose a stack name and fill in the template URL with: https://nuxeo.s3.amazonaws.com/templates/Nuxeo.template.


Create Stack

Cancel

SELECT TEMPLATE SPECIFY PARAMETERS REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of you own templates stored in S3 or on your local hard drive.

Stack Name: MyVeryOwnNuxeo

Stack Template Source:

- Use a sample template
- Upload a Template File
- [✓ Provide a Template URL]

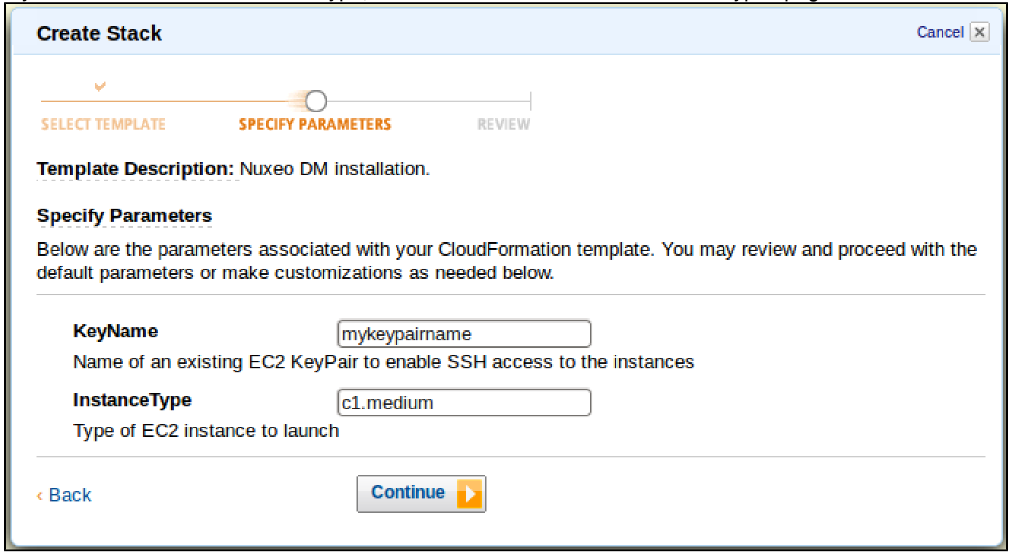
https://nuxeo.s3.amazonaws.com/templates/NuxeoDM.template

Show Advanced Options

Continue

Note: in some regions, AWS will tell you that is not a S3 URL, in that case just download the file locally and use the "Upload a template" option.

5. Fill in your previously created keypair name (KeyName) and the type of amazon instance you want.
   You can find a list of instance types at http://aws.amazon.com/ec2/instance-types/. The default (c1.medium) is suitable for small to medium size installations.
   If you choose a different instance type, check its “API name” on the instance types page and use that for the "InstanceType" field.


Create Stack

Cancel

SELECT TEMPLATE SPECIFY PARAMETERS REVIEW

Template Description: Nuxeo DM installation.

Specify Parameters

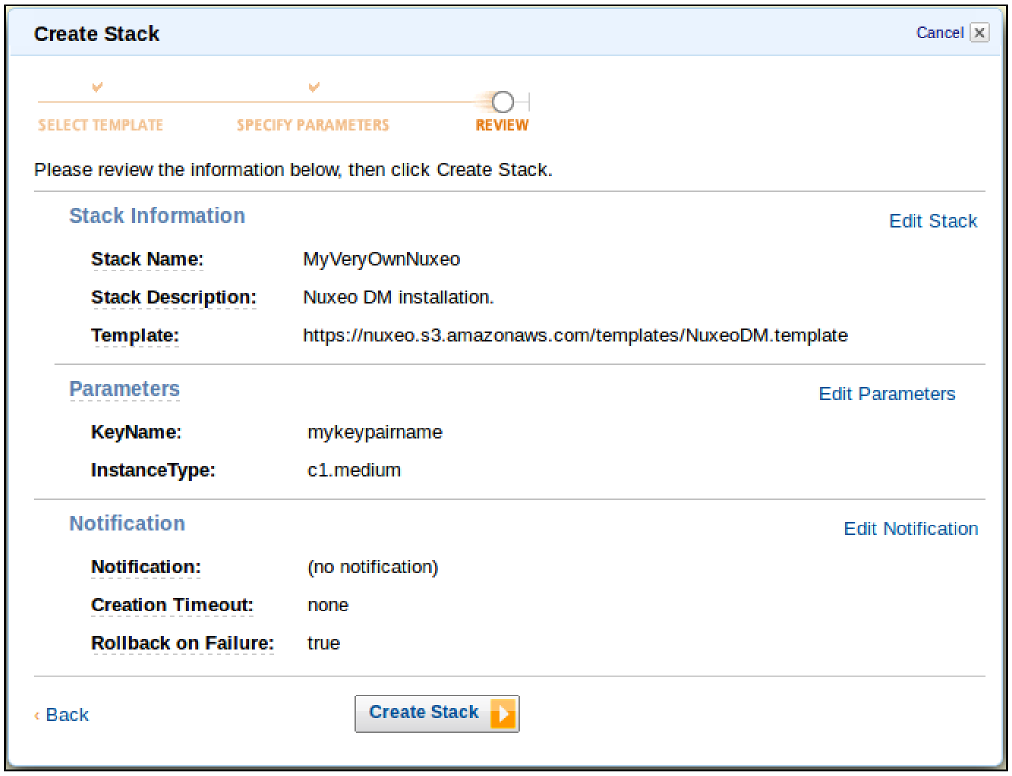
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

KeyName: mykeypairname
Name of an existing EC2 KeyPair to enable SSH access to the instances

InstanceType: c1.medium
Type of EC2 instance to launch

‹ Back Continue

6. Review your settings and click on the "Create Stack" button to start the creation process.



Create Stack

Cancel

SELECT TEMPLATE — SPECIFY PARAMETERS — REVIEW

Please review the information below, then click Create Stack.

Stack Information — Edit Stack

Stack Name:	MyVeryOwnNuxeo
Stack Description:	Nuxeo DM installation.
Template:	https://nuxeo.s3.amazonaws.com/templates/NuxeoDM.template

Parameters — Edit Parameters

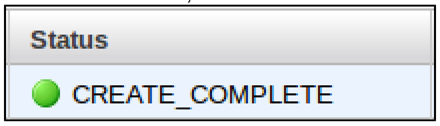
KeyName:	mykeypairname
InstanceType:	c1.medium

Notification — Edit Notification

Notification:	(no notification)
Creation Timeout:	none
Rollback on Failure:	true

‹ Back — Create Stack

After a few minutes, the instance creation is complete.



Status

● CREATE_COMPLETE

Hit the “Refresh” button on the top right corner of the page now and then as it doesn't auto-refresh.

7. Select the line that shows your new CloudFormation stack, then the “Outputs” tab at the bottom.
   It shows the URL at which you can reach your brand new Nuxeo.

Note that it can take a few more minutes for Nuxeo to be active on that URL as there can still be some installation tasks running at this point.

The template can be used for testing and production purposes. As for every production setup, you will want to check that the configuration suits your needs and tune it as needed: HTTPS setup, disk size, ...

Deploying Nuxeo on CloudFoundry

Nuxeo needs JAXB libraries more recent than what's available in Java 6, which means that to deploy Nuxeo we must:

- Either use Java 6 with endorsed JARs,
- Or use Java 7.

As it's not possible in CloudFoundry (in Spring mode, which is the one allowing the deployment of WAR files) to use an endorsed directory, the only option is to use Java 7.

Building a WAR

The following is a recipe to build a WAR of Nuxeo CoreServer. We need a WAR because normally Nuxeo runs as an exploded hierarchy with a customized loader that knows about the Nuxeo structure. Building a simple WAR turns this into a static configuration that can be used in any servlet container.

Getting Nuxeo

Download the latest nuxeo-coreserver from http://qa.nuxeo.org/jenkins/view/Depl/job/IT-nuxeo-master-build/ , for instance in these examples: `nuxeo-coreserver-5.7-I20121123_0116-tomcat.zip`

For this recipe we'll put it into `/opt`.

In this section

Enabling Tomcat 7

Nuxeo is currently packaged with a Tomcat 6 instance. The following allows it to work in a Tomcat 7 instance.

1. First get a Tomcat 7 instance and unzip it into /opt/tomcat-7.0.30

```
cd /opt
unzip apache-tomcat-7.0.30.zip
mv apache-tomcat-7.0.30 tomcat-7.0.30
```

2. Then convert Nuxeo to work in that Tomcat 7 instance:

```
cd /opt
unzip nuxeo-coreserver-5.7-I20121123_0116-tomcat.zip
T6=nuxeo-coreserver-5.7-I20121123_0116-tomcat
T7=tomcat-7.0.30
rm -rf $T7/nxserver
rm -rf $T7/templates
rm -rf $T7/endorsed
cp -R $T6/nxserver $T7/
cp -R $T6/templates $T7/
cp -R $T6/endorsed $T7/
cp $T6/bin/nuxeoctl $T7/bin/
cp $T6/bin/nuxeo.conf $T7/bin/
cp $T6/bin/nuxeo-launcher.jar $T7/bin/
cp $T6/lib/log4j.xml $T7/lib/
cp $T6/lib/nuxeo-*.jar $T7/lib/
cp $T6/lib/commons-logging-*.jar $T7/lib/
cp $T6/lib/commons-lang-*.jar $T7/lib/
cp $T6/lib/freemarker-*.jar $T7/lib/
cp $T6/lib/log4j-*.jar $T7/lib/
cp $T6/lib/lucene-*.jar $T7/lib/
cp $T6/lib/mail-*.jar $T7/lib/
cp $T6/templates/tomcat7/conf/server.xml.nxftl
$T7/templates/common-base/conf/server.xml.nxftl
chmod +x $T7/bin/*.sh
```

3. In `bin/nuxeo.conf` , remove or comment `nuxeo.wizard.done=false.`

Building the Tomcat+WAR

This builds a ZIP containing a WAR and a few additional files, designed to be unpacked at the root of a Tomcat instance. It cannot be used directly as a pure WAR.

```
cd /opt/tomcat-7.0.30
bin/nuxeoctl pack /opt/nuxeotomcatwar.zip
```

The resulting ZIP is all that's needed for the rest of the instructions.

Preparing a full Tomcat 7 with Nuxeo WAR

1. Set up a new Tomcat 7 from scratch with the WAR that was just built.

   ```
   cd /opt
   rm -rf tomcat-7.0.30
   unzip apache-tomcat-7.0.30.zip
   mv apache-tomcat-7.0.30 tomcat-7.0.30
   cd tomcat-7.0.30
   chmod +x bin/*.sh
   unzip /opt/nuxeotomcatwar.zip
   ```

2. Now update the WAR to work in a Tomcat 7 instance without references to the toplevel `>lib/` directory:

   ```
   mv lib/commons-*.jar lib/h2-*.jar lib/lucene-*.jar lib/nuxeo-*.jar
   webapps/nuxeo/WEB-INF/lib/
   rm lib/derby-*.jar
   ```

3. Finally we must set up the Nuxeo datasources inside the WAR (instead of from the toplevel `server.xml` in a standard Nuxeo setup, which is more convenient when possible).

   Edit the file `webapps/nuxeo/META-INF/context.xml`. In this file the `<Resource>` elements that point to a global configuration in `server.xml` must be replaced by explicit resources:

```
  <Resource name="jdbc/NuxeoDS" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
  <Resource name="jdbc/nxsqldirectory" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
  <Resource name="jdbc/nxrelations-default-jena"
accessToUnderlyingConnectionAllowed="true" auth="Container"
driverClassName="org.h2.Driver" maxActive="100" maxIdle="30" maxWait="10000"
password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo" username="sa"
validationQuery=""/>
  <Resource name="jdbc/comment-relations"
accessToUnderlyingConnectionAllowed="true" auth="Container"
driverClassName="org.h2.Driver" maxActive="100" maxIdle="30" maxWait="10000"
password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo" username="sa"
validationQuery=""/>
  <Resource name="jdbc/nxaudit-logs" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
  <Resource name="jdbc/nxjbpm" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
  <Resource name="jdbc/placeful_service_ds"
accessToUnderlyingConnectionAllowed="true" auth="Container"
driverClassName="org.h2.Driver" maxActive="100" maxIdle="30" maxWait="10000"
password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo" username="sa"
validationQuery=""/>
  <Resource name="jdbc/nxwebwidgets" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
  <Resource name="jdbc/nxuidsequencer" accessToUnderlyingConnectionAllowed="true"
auth="Container" driverClassName="org.h2.Driver" maxActive="100" maxIdle="30"
maxWait="10000" password="" type="javax.sql.DataSource" url="jdbc:h2:nuxeo"
username="sa" validationQuery=""/>
```

In the above we use an embedded H2 database. Configuration for an external database server would need to use a different JDBC URL.

4. The above exploded WAR can now be packed into a zipped WAR:

```
cd /opt/tomcat-7.0.30/webapps/nuxeo
zip -r /opt/nuxeo.war .
```

The resulting `/opt/nuxeo.war` is now compatible with any modern application server.

Deploying to CloudFoundry

Patching the WAR

If you try to deploy the above WAR on CloudFoundry you'll get an error: `Too many open files`

This is due to the fact that CloudFoundry has a limit of 256 file descriptors open, and Nuxeo uses more.

In fact, it's the Tomcat classloader that uses a lot of file descriptors, because for performance reasons it keeps one open file descriptor per JAR in the `WEB-INF/lib` directory. To work around this problem, the non-Nuxeo JARs present in `WEB-INF/lib` can be merged into one big JAR, which solves the problem.

```
cd webapps/nuxeo/WEB-INF/lib
mkdir tmp
mkdir bigjar
mv *.jar tmp/
mv tmp/nuxeo-*.jar .
cd bigjar
for i in ../tmp/*.jar; do unzip $i; done
rm META-INF/*.SF META-INF/*.DSA
zip -r ../bigjar.jar .
cd ..
rm -rf tmp/ bigjar/
```

Deploying Tomcat 7 + Nuxeo

Before the above Tomcat 7 instance can be set up as a full “standalone” application in CloudFoundry, it needs to be modified to take configuration information (like TCP ports) from the CloudFoundry PaaS framework. (These are standard instructions for using a stock Tomcat in CloudFoundry, you'll find them elsewhere.)

1. The following is added to `bin/catalina.sh` :

   ```
   # USE VCAP PORT IF IT EXISTS, OTHERWISE DEFAULT TO 8080
   if [ -z ${VCAP_APP_PORT} ]; then
     export VCAP_APP_PORT=8080
   fi
   export JAVA_OPTS="-Dport.http.nonssl=$VCAP_APP_PORT $JAVA_OPTS"
   ```

2. The `bin/startup.sh` is changed to not run in the background (`run` instead of `start`):

   ```
   exec "$PRGDIR"/"$EXECUTABLE" run "$@"
   ```

3. Finally the non-useful ports in `conf/server.xml` are commented out:
   - Replace 8005 with -1 to deactivate the Tomcat SHUTDOWN port,
   - Comment out the AJP connector on port 8009,
   - Replace 8080 with the expression `${port.http.nonssl}` to use the PaaS configuration.
4. The Tomcat 7 application including the Nuxeo WAR is now ready to be pushed to CloudFoundry as a “Standalone Application”. For this, standard CloudFoundry mechanisms are used, don't forget to specify the Java 7 runtime:

   ```
   vmc push --runtime=java7 myapp
   ```

   The following CloudFoundry YML manifest corresponds to a full setup (`myapp` should be replaced by your namespace):

```
---
applications:
  .:
    mem: 1G
    instances: 1
    url: myapp.cloudfoundry.com
    framework:
      info:
        mem: 64M
        exec:
        description: Standalone Application
      name: standalone
    command: bin/startup.sh
    name: myapp
    runtime: java7
```

Testing Nuxeo CMIS

Once deployed and started, Nuxeo CoreServer does not provide a web-accessible graphical user interface (because the CoreServer version doesn't have those), but it can be addressed through a CMIS client like the Apache CMIS Workbench available at http://chemistry.apache.org/java/developing/tools/dev-tools-workbench.html .

It must point to Nuxeo, whose CMIS address is described by the Nuxeo startup page, usually it is of the form http://localhost:8080/nuxeo/atom/cmis . The default Nuxeo user/password is Administrator/Administrator.

How to Estimate Disk and Database Usage

Documents in Nuxeo are more than "simple files". The files are called "binaries" or "blobs" (even if they are text files).

Volume usage depends on the type of documents but it is possible to estimate the required space on filesystem and database:

- Filesystem
  - Base unit will be the size of the binaries.
  - If there are images, then you must take in account the generated thumbnails: depending on the original image size, the thumbnail will be about 10% to 100% of the original size. Let's consider an average of 50%.
  - Backup: it's possible to perform remote backup (rsync for instance) but if you plan to use a local backup, then the required size must be doubled.
  - Software
    - Disk usage by Nuxeo is stable and about 300MB.
    - Temporary files (when uploading for instance): reserve some space which depends on the maximum size of imported files (the temporary directory can be configured).
    - Logs: based on Log4J, the log files can be easily configured (size limit, rolling rules, ...).
- Database
  - Binaries x2: if binaries are text files, then you can estimate to twice of the binaries size in database for storing their "fulltext" indexes.
  - +10% of binaries: other metadata usually consumes about 10% of the binaries total size.
  - Backup: unless using backup streaming (not available for all databases and complex to setup), you will need a local backup so you must double the available size versus the estimated usage.

The easiest way is still to use the application for some duration, get usage statistics and deduce the required size.

So, based on the above information, if you have X GB of documents of which Y GB of images,

- Filesystem
  - data = X + 50% Y (x2 for backup).
  - /tmp = 1GB + maximum size of imported documents
  - /var/log = 50MB~5GB depending on the Log4J rules and the log level (error/warn/info/debug/trace).
  - application = ~1GB
- Database = 210% X (x2 for backup).

Setup

The Nuxeo Platform provides you with easy access to the configuration of your Nuxeo server, thanks to the Admin Center and the Startup Wizard. For advanced configuration or a simple review, manual edition of Nuxeo's configuration file, called `nuxeo.conf`, and a template system is also available.

Initial Setup of the Nuxeo Platform with the Startup Wizard

The first time you start the Nuxeo Platform and go the URL http://localhost:8080/nuxeo, a Startup Wizard will guide you to the main configuration steps and enable you to choose which modules you want to enable on the Platform. For each step, a default setting is proposed that enables you to test the application. You can change this default configuration to adapt it to specific environments.

The settings defined during the initial setup can be changed afterward using the Admin Center or by editing Nuxeo's configuration file manually. Modules can also be added or removed afterwards from the Admin Center.

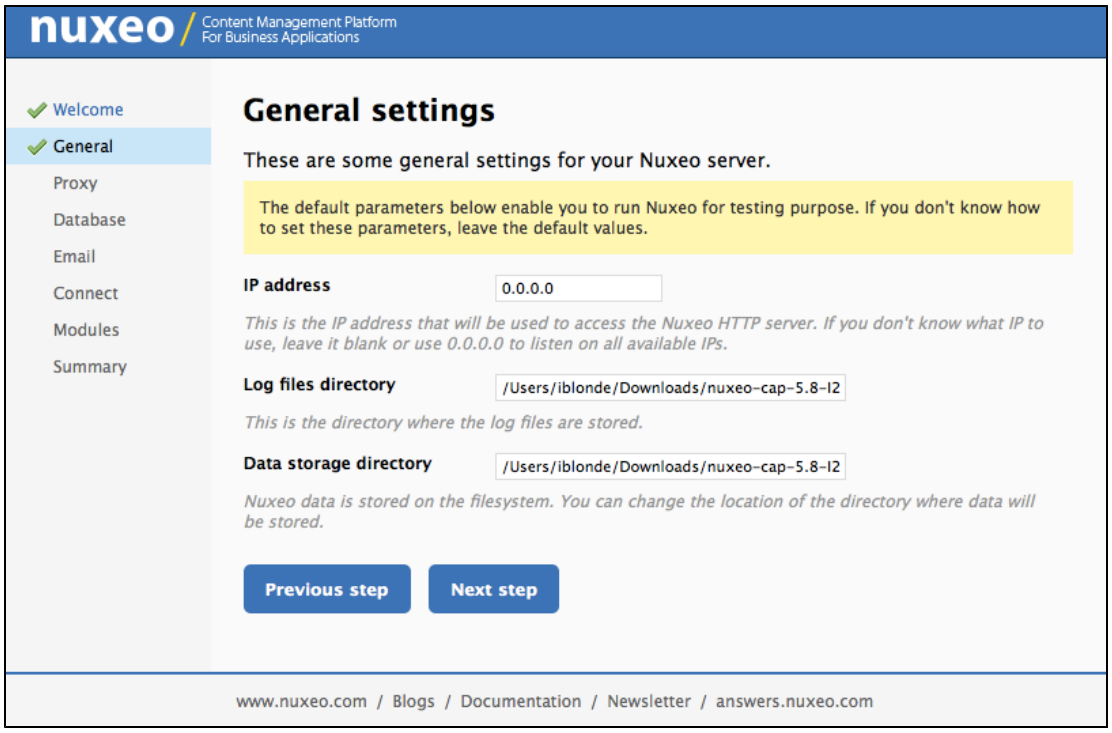
The Startup wizard will be run only if the configuration sets `nuxeo.wizard.done=false`. You can edit the value in order to replay the wizard (using the Admin Center or editing the nuxeo.conf file manually), or simply run `nuxeoctl wizard`

For Internet Explorer 9 users
You need to add the Nuxeo server URL in the trusted sites list to be able to complete the installation and configuration steps.
In the Internet Options > Security > Trusted Sites menu, click on the **Sites** button, type the Nuxeo server URL and add it.

On this page

Server General Settings

This step enables you to change the default IP address of the server and where the logs and data are stored.


nuxeo/ Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Modules
Summary
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

General settings

These are some general settings for your Nuxeo server.

The default parameters below enable you to run Nuxeo for testing purpose. If you don't know how to set these parameters, leave the default values.

IP address: 0.0.0.0

This is the IP address that will be used to access the Nuxeo HTTP server. If you don't know what IP to use, leave it blank or use 0.0.0.0 to listen on all available IPs.

Log files directory: /Users/iblonde/Downloads/nuxeo-cap-5.8-I2

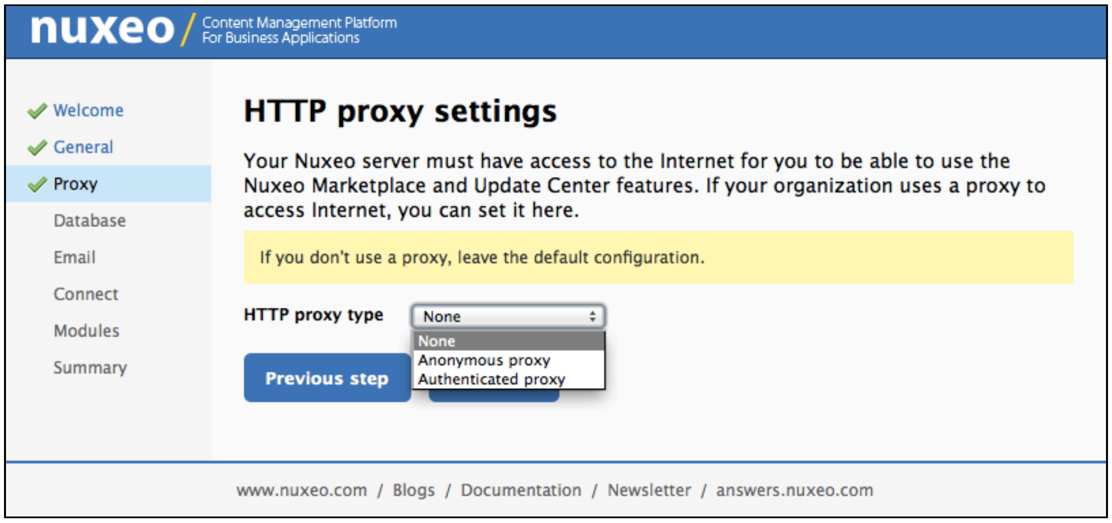
This is the directory where the log files are stored.

Data storage directory: /Users/iblonde/Downloads/nuxeo-cap-5.8-I2

Nuxeo data is stored on the filesystem. You can change the location of the directory where data will be stored.

Proxy Settings

Some features of Nuxeo applications requires to access the Internet. That's the case of the Update Center from which you can access to the Marketplace add-ons and plugins, updates for your application, your Studio customizations.


Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Modules
Summary
None
Anonymous proxy
Authenticated proxy
Previous step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

HTTP proxy settings

Your Nuxeo server must have access to the Internet for you to be able to use the Nuxeo Marketplace and Update Center features. If your organization uses a proxy to access Internet, you can set it here.

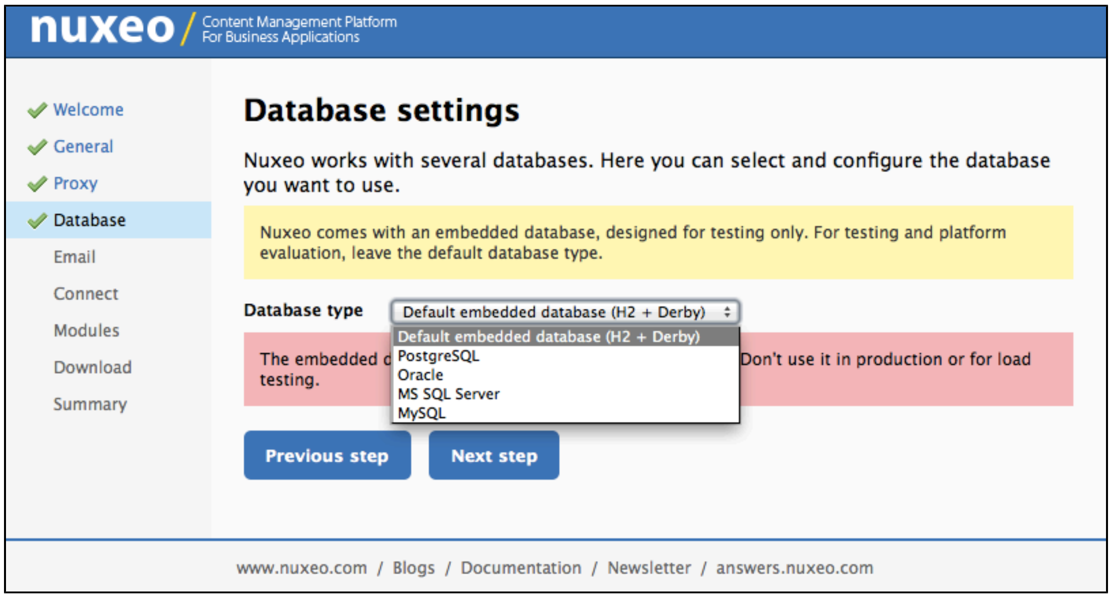
If you don't use a proxy, leave the default configuration.

HTTP proxy type: None

Database Settings

Nuxeo applications embed a database by default, called H2/Derby. This database enables you to fully test and evaluate the application. However it is not recommended to use this embedded database for production and load testing. Select the database you want to use and provide the connection information to the database.
Possible databases are:

- PostgreSQL,
- Oracle,
- MS SQL Server,
- MySQL.


Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Modules
Download
Summary
Default embedded database (H2 + Derby)
PostgreSQL
Oracle
MS SQL Server
MySQL
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

Database settings

Nuxeo works with several databases. Here you can select and configure the database you want to use.

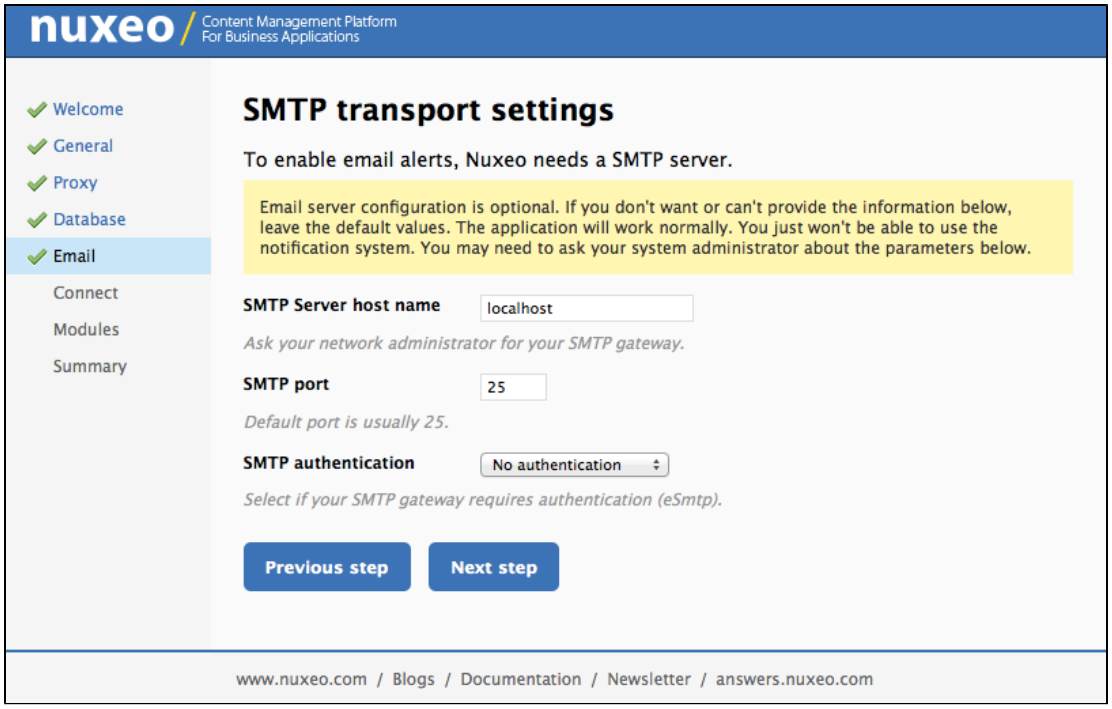
Nuxeo comes with an embedded database, designed for testing only. For testing and platform evaluation, leave the default database type.

Database type: Default embedded database (H2 + Derby)

The embedded d Don't use it in production or for load testing.

Mail Settings

Nuxeo applications include email alert features. By default, no SMTP configuration is enabled and therefore no email alerts will be sent to users. You can refer to the email alerts section for more information about the SMTP configuration.


nuxeo/ Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Modules
Summary
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

SMTP transport settings

To enable email alerts, Nuxeo needs a SMTP server.

Email server configuration is optional. If you don't want or can't provide the information below, leave the default values. The application will work normally. You just won't be able to use the notification system. You may need to ask your system administrator about the parameters below.

SMTP Server host name: localhost

Ask your network administrator for your SMTP gateway.

SMTP port: 25

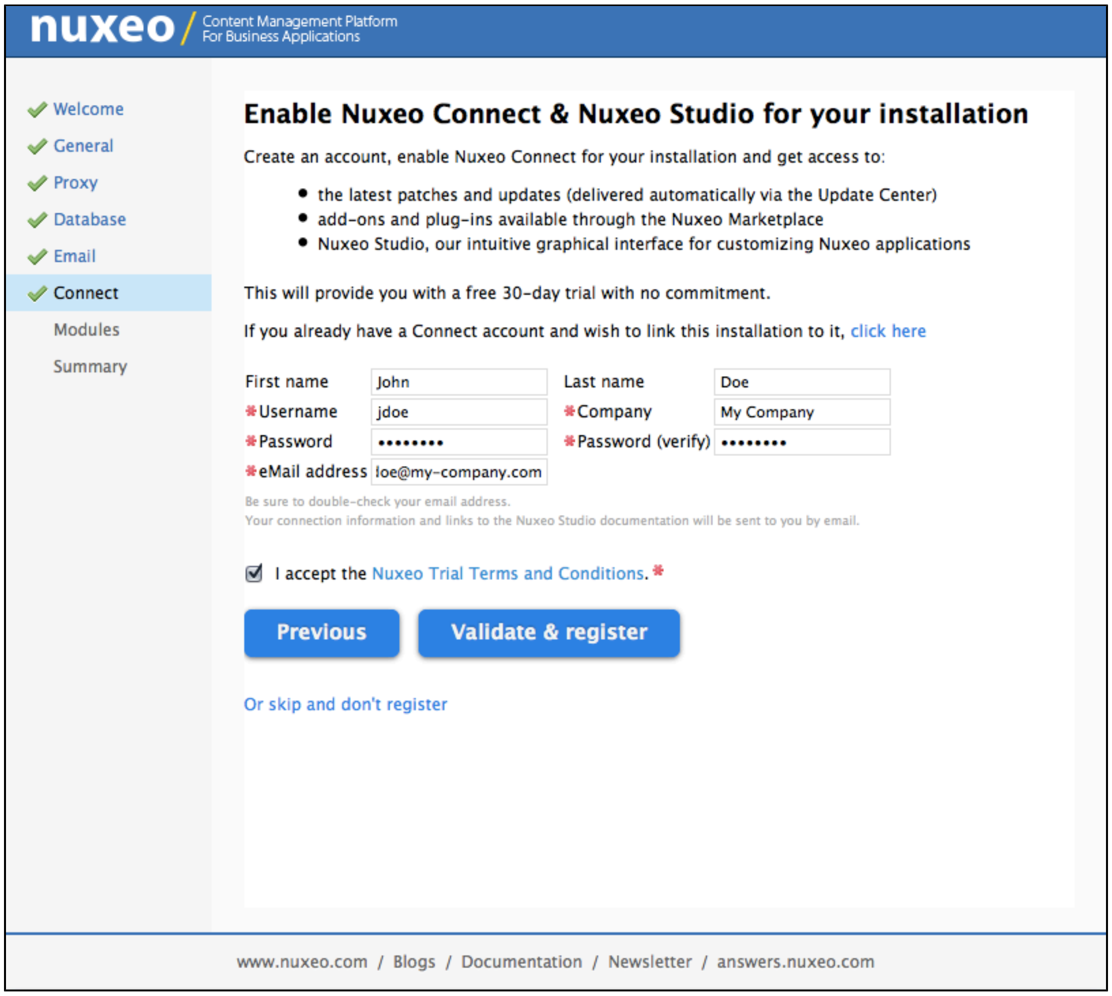
Default port is usually 25.

SMTP authentication: No authentication

Select if your SMTP gateway requires authentication (eSmtp).

Connect Settings

From this step, you can subscribe to a free 30 days trial offer of Nuxeo Connect which gives you the possibility to evaluate and fully leverage the Marketplace catalog and Nuxeo Studio, the online Nuxeo customization environment. If you subscribe to the trial offer of Nuxeo Connect, you will be sent an email confirming your subscription and your credentials to Nuxeo Connect and giving you the links to access the Nuxeo Connect Portal and Nuxeo Studio.


nuxeo / Content Management Platform For Business Applications
✓ Welcome
✓ General
✓ Proxy
✓ Database
✓ Email
✓ Connect
Modules
Summary
Previous
Validate & register
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

Enable Nuxeo Connect & Nuxeo Studio for your installation

Create an account, enable Nuxeo Connect for your installation and get access to:

- the latest patches and updates (delivered automatically via the Update Center)
- add-ons and plug-ins available through the Nuxeo Marketplace
- Nuxeo Studio, our intuitive graphical interface for customizing Nuxeo applications

This will provide you with a free 30-day trial with no commitment.

If you already have a Connect account and wish to link this installation to it, click here

First name	John	Last name	Doe
*Username	jdoe	*Company	My Company
*Password	••••••••	*Password (verify)	••••••••
*eMail address	loe@my-company.com		

Be sure to double-check your email address.
Your connection information and links to the Nuxeo Studio documentation will be sent to you by email.

☑ I accept the Nuxeo Trial Terms and Conditions. *

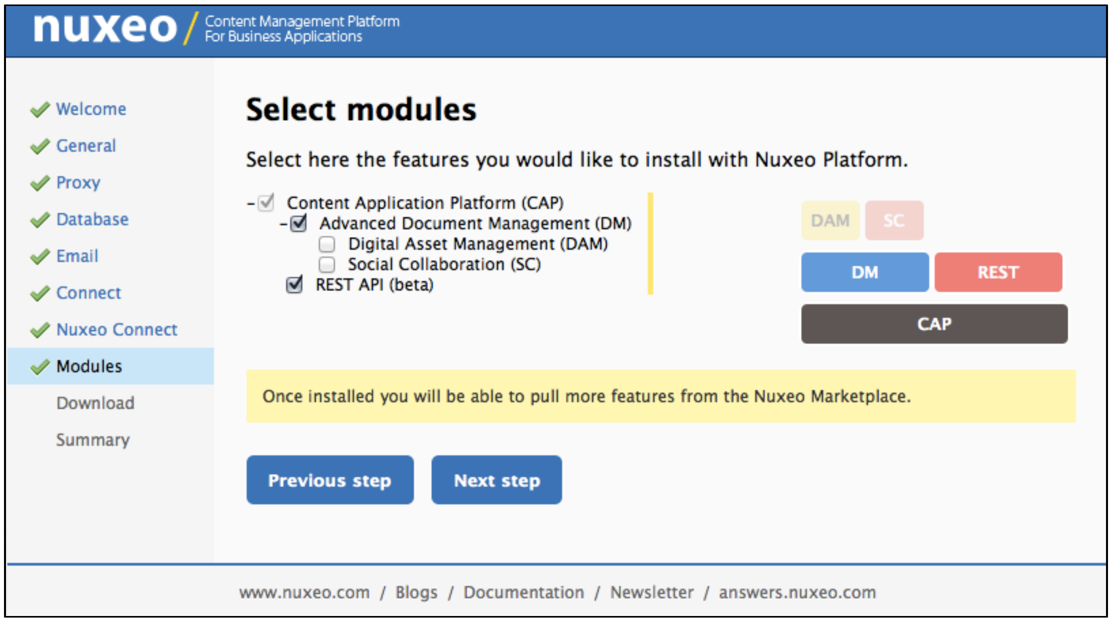
Or skip and don't register

If you already have a Nuxeo Connect account, you can register your Nuxeo instance from this step to directly be able to apply your Nuxeo Studio customizations and the installation of Nuxeo Marketplace packages in your instance.

Module Installation

Select the modules you want to install on the Platform. You can also just keep the naked Content Application Platform.

You can install or uninstall modules afterwards from the Admin Center.


nuxeo / Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Nuxeo Connect
Modules
Download
Summary
DAM
SC
DM
REST
CAP
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

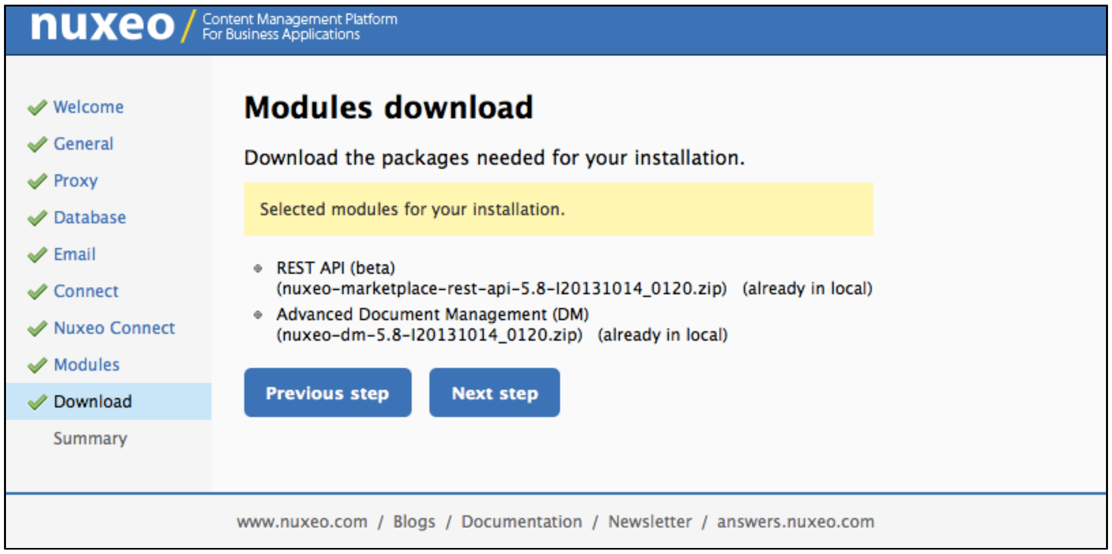
Select modules

Select here the features you would like to install with Nuxeo Platform.

- ✓ Content Application Platform (CAP)
  - ✓ Advanced Document Management (DM)
    - ☐ Digital Asset Management (DAM)
    - ☐ Social Collaboration (SC)
  - ✓ REST API (beta)

Once installed you will be able to pull more features from the Nuxeo Marketplace.

And if needed, download the module packages. Packages may be already included in the Platform.


nuxeo / Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Nuxeo Connect
Modules
Download
Summary
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

Modules download

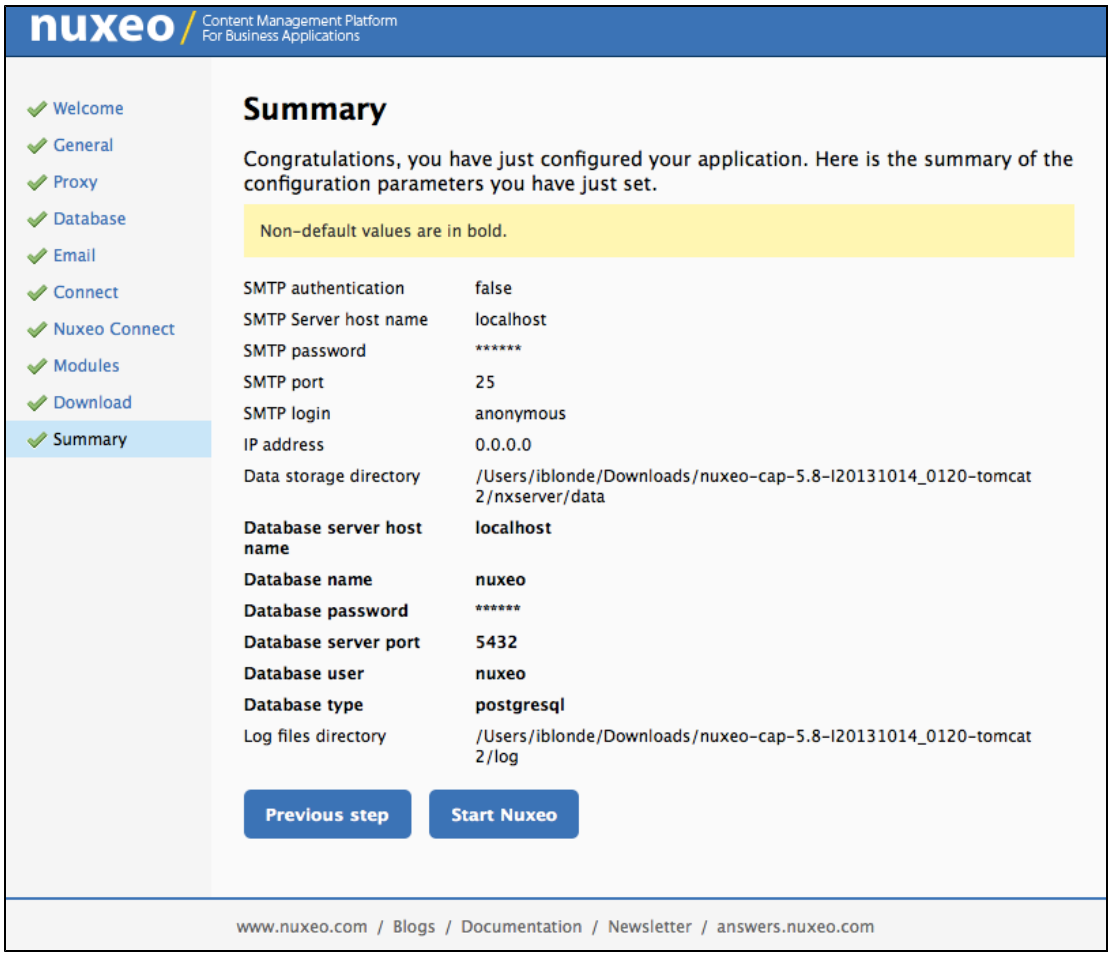
Download the packages needed for your installation.

Selected modules for your installation.

- REST API (beta)
  (nuxeo-marketplace-rest-api-5.8-I20131014_0120.zip) (already in local)
- Advanced Document Management (DM)
  (nuxeo-dm-5.8-I20131014_0120.zip) (already in local)

Summary

A final Summary step provides you with a screen on which you can see all the configuration parameters that you set in the previous steps so you can review them and possibly go back to a step to change them.


Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Nuxeo Connect
Modules
Download
Summary
Previous step
Start Nuxeo
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

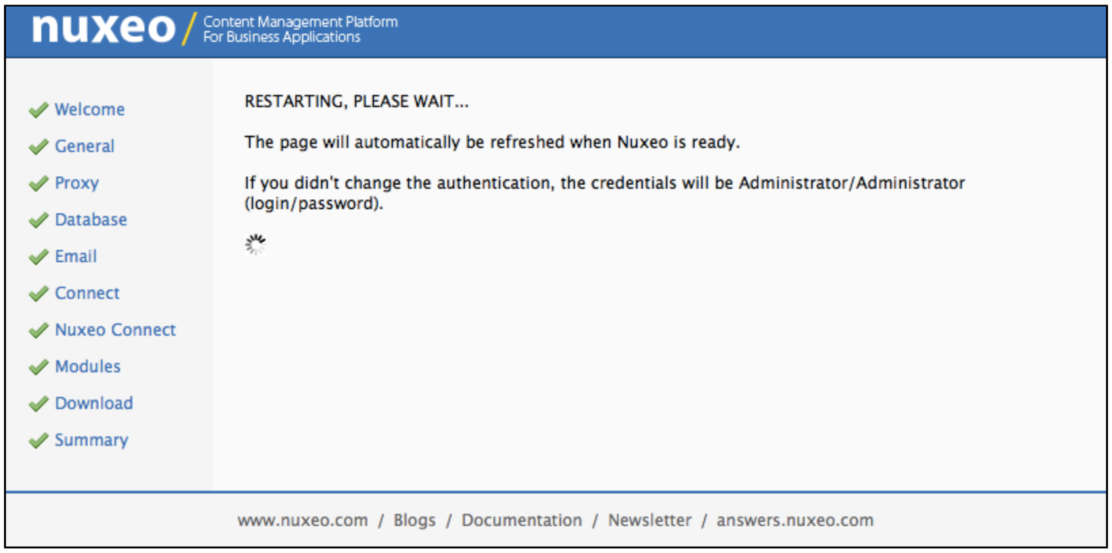
Summary

Congratulations, you have just configured your application. Here is the summary of the configuration parameters you have just set.

Non-default values are in bold.

SMTP authentication	false
SMTP Server host name	localhost
SMTP password	******
SMTP port	25
SMTP login	anonymous
IP address	0.0.0.0
Data storage directory	/Users/iblonde/Downloads/nuxeo-cap-5.8-I20131014_0120-tomcat2/nxserver/data
Database server host name	**localhost**
Database name	**nuxeo**
Database password	******
Database server port	**5432**
Database user	**nuxeo**
Database type	**postgresql**
Log files directory	/Users/iblonde/Downloads/nuxeo-cap-5.8-I20131014_0120-tomcat2/log

To validate your configuration, click on the **Start Nuxeo** button. The server will automatically restart and your configuration will be applied. Once the server is restarted, you are displayed the login page. Log in to your application the **Administrator** user name and the **Administrator** passwor d.


Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Nuxeo Connect
Modules
Download
Summary
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

RESTARTING, PLEASE WAIT...

The page will automatically be refreshed when Nuxeo is ready.

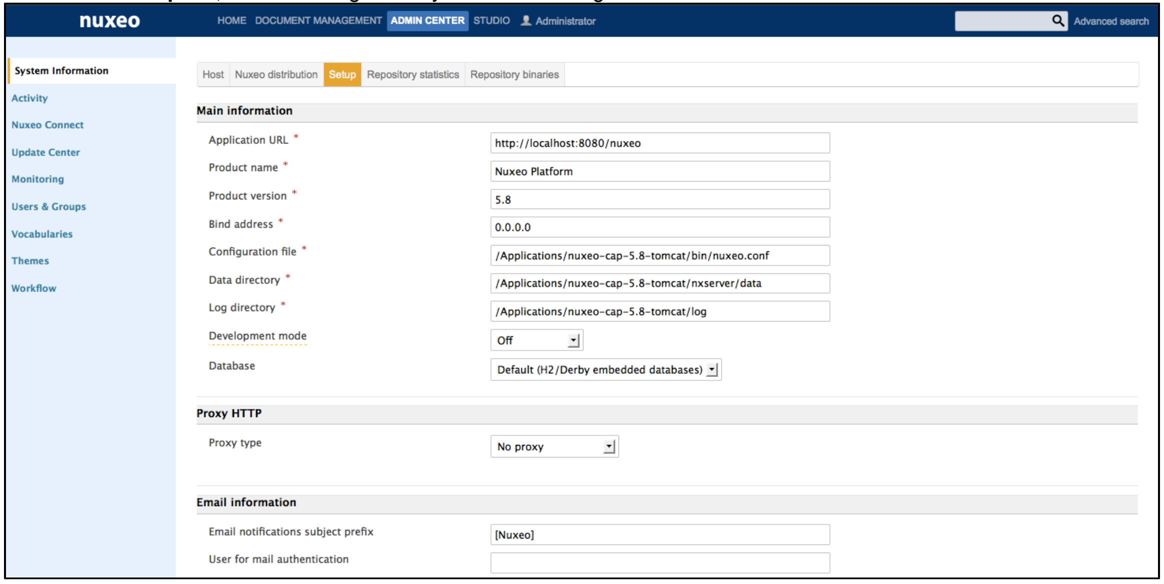
If you didn't change the authentication, the credentials will be Administrator/Administrator (login/password).

Update the Application's Configuration Using the Admin Center

The Admin Center is the graphical interface that enables the application's administrators to edit the configuration of the application directly from the user interface, and prevents them from editing .xml and .conf files. They can edit the configuration of the application, monitor it, display messages to the users, and easily customize the application thanks to the Update Center.

To edit the configuration of the application using the Admin Center:

1. Log in with an administrator account.
   Default administrator credentials are:
   - login: Administrator
   - password: Administrator
2. Click on the **Nuxeo Admin Center** link in the page header.
3. Click on the **Setup** tab, edit the configuration you want to change and Save.



Main information

Application URL *	http://localhost:8080/nuxeo
Product name *	Nuxeo Platform
Product version *	5.8
Bind address *	0.0.0.0
Configuration file *	/Applications/nuxeo-cap-5.8-tomcat/bin/nuxeo.conf
Data directory *	/Applications/nuxeo-cap-5.8-tomcat/nxserver/data
Log directory *	/Applications/nuxeo-cap-5.8-tomcat/log
Development mode	Off
Database	Default (H2/Derby embedded databases)

Proxy HTTP

Proxy type	No proxy

Email information

Email notifications subject prefix	[Nuxeo]
User for mail authentication	

4. If indicated as needed on top of the page, restart the server.

You can also take a look at the following pages for recommendations and examples:

- Recommended Configurations,
- Configuration Examples.

You can report to the Configuration Parameters Index (nuxeo.conf) for more information about the available parameters.

Manual Edition of Nuxeo Configuration File `nuxeo.conf`

By default, the `nuxeo.conf` file is located in `$NUXEO_HOME/bin`. If you installed your application using the Windows installer, the configuration is located in `%APPDATA%\Nuxeo DM\conf` (check the corresponding Knowledge Base page for more information). If you plan to use the application in production, you should move the configuration file outside the Nuxeo home directory, to make upgrades easier and more secured: your data and configuration won't risk to be overridden or lost.

For Windows users
Do not use Office writers, nor Notepad.

Wordpad is fine, Notepad+\+ and SciTE are good text editors, there are a lot of other text editors.

You can report to the Configuration Parameters Index (nuxeo.conf) for the list of available paramaters.

Related content

- Setup
- Admin Center overview
- Updating your Instance with Studio Configuration
- Configuration Templates
- Uninstalling a Package
- Registering your Nuxeo Instance
- Installing a New Package on Your Instance
- Configuration Examples
- Recommended Configurations
- Nuxeo Clustering Configuration

Configuration Parameters Index (nuxeo.conf)

Recommended Configurations

Nuxeo applications come as ready-to-use applications, that you can quickly install and evaluate. However, if you plan to go in production, have several Nuxeo applications on the same machine or do some performance tests, here are some changes of configuration that we recommend to do, especially for advanced testing or before going into production:

The steps given below are given using the Admin Center. They can of course also be done by editing the `nuxeo.conf` file manually.

More configuration use cases on the Configuration Examples page.

Moving Configuration, data and log Directories Outside Nuxeo

The configuration of your application is saved in the `nuxeo.conf` configuration file, whatever the means you use to configure your application (manual edit, Startup Wizard or Admin Center). It is better, although not mandatory, to store your customized configuration outside Nuxeo. This way, you will be able to easily upgrade Nuxeo, keeping your configuration safely apart of Nuxeo directory.

On this page

To move the configuration file outside the Nuxeo directory:

1. Move the `nuxeo.conf` file from its default location.
2. After you moved `nuxeo.conf`, you need to define its location as an environment variable.

By default, `data` and `log` directories are stored inside the Nuxeo tree. To ease backup and upgrades, it is highly recommended to move them outside the Nuxeo tree.

To move the data and log directories:

1. In the Admin Center's **System Information** > **Setup** tab, type the path to the location where you want the directories to be stored (see the table below).
2. Click on **Save**.
3. Restart your server.
   The `data` and `log` directories are created at the location you typed.

Data and log directories configuration

Field / Property	Description
Data directory `nuxeo.data.dir`	Data directory (absolute or relative to NUXEO_HOME). It involves all data not being stored in the database. Linux recommended path: `/var/lib/nuxeo/...`
Log directory `nuxeo.log.dir`	Log directory (absolute or relative to NUXEO_HOME). Linux recommended path: `/var/log/nuxeo/...`

Defining Environment Variables

When the server starts, it guesses where the Nuxeo home directory and the Nuxeo configuration file (`nuxeo.conf`) are located. If it doesn't find it or if you want to force it to use a specific home directory and/or a specific configuration file, you can define their location as environment variables.

`NUXEO_HOME`

Here is how Nuxeo home is guessed when the server starts: If `NUXEO_HOME` is not set, then the parent directory of the called script (`nuxeoctl`) is used.

Setting the Nuxeo home directory as an environment variable is recommended in the following cases:

- if you installed several Nuxeo applications on the same machine (for evaluation or production purpose),
- if you want to use other scripts than the `$NUXEO_HOME/bin/nuxeoctl` script (such as a service in `/ect/init.d`).

You must then set `NUXEO_HOME=/path/to/nuxeo/` in the system environment variables:

- Windows users must write `"set NUXEO_HOME=..."` or use the control panel interface to define user environment parameters (like it's done for `%PATH%`).
- Linux and Mac OS X users will write `"export NUXEO_HOME=...."` in `~/.bashrc` or `~/.profile`.

`NUXEO_CONF`

You need to set the location of the `nuxeo.conf` file as an environment variable if you moved your configuration outside of the Nuxeo directory.

Moving the data and configuration outside the Nuxeo directory is recommended in a production environment because it makes upgrades easier and more secured: Your data and configuration won't risk to be overridden or lost. You must then set `NUXEO_CONF=/path/to/nuxeo.conf` in the system environment variables.

Windows Specific Case

Under Windows, the location of the `nuxeo.conf` is defined by that order of priority (i.e. first one of those found is used):

- Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\%PRODNAME%\ConfigFile` with `%PRODNAME%` equals to "Nuxeo" (or in older versions, "Nuxeo CAP", "Nuxeo DM", "Nuxeo DAM", ...),
- Environment variable `NUXEO_CONF`,
- `"nuxeo.conf"` file in the working directory,
- `"nuxeo.conf"` file on the Desktop,
- `"nuxeo.conf"` file in the same location as `nuxeoctl.bat`.

Changing the Default Embedded Database

The Nuxeo Platform provides a default embedded database for testing and evaluation purpose, called Derby. However, it is not adapted for a production environment (see the H2 limitations page).

Before going live, you should configure one of the production-safe database supported by Nuxeo. See the Database section of this documentation.

Enabling e-Mail Alerts

The default Nuxeo Platform email configuration is filled in with neutral values that you need to edit to make the platform work with your mail server. Unless you do that, alerts emails won't be sent to users.

To make alerts available:

1. In the Admin Center, click on the **Setup** tab of System Information section.
2. Edit and fill in the values of the Email information section (see below for expected parameters).

   ✓ To enable alerts, filling in the SMTP parameters should be sufficient for most mail server configurations.
3. Click the button **Save**.
   As indicated on top of the page, you need to restart your server so the new configuration is taken into account.

Email information configuration

Field / Property	Description

Email notifications subject prefix `nuxeo.notification.eMailSubjectPrefix`	Text displayed in the "Object" before the object of the alerts email to help users identify that the emails are coming from the application. Default value is "[Nuxeo]". You can change is to whatever value you like.
Mail store protocol `mail.store.protocol`	Name of the protocol used to store emails on the server. Default value is "pop3". You may need to change it to "IMAP".
Mail transport protocol `mail.transport.protocol`	Name of the protocol used to send emails. Default value is "smtp". This should work in most cases.
Host name for POP3 `mail.pop3.host`	Name of the mail server host used to receive and store emails. Default value is "pop3.nosuchhost.nosuchdomain.com". You need to change it.
Debug mode `mail.debug`	Default value is set to "false". Change it to "true" if you want to have the details of what the server is doing in the logs.
Host name for SMTP `mail.smtp.host`	Mail server host name for outgoing mails. Default value is "localhost". You need to change it so emails can be sent from the server.
Port number for SMTP `mail.smtp.port`	Mail server port for outgoing emails. Default value is 25.
Use authentication for SMTP `mail.smtp.auth`	Indicate if authentication is needed for the mail server to send emails. Default value is "true". You should change it to "false" if no authentication for sending email is required.
Use STARTTLS for SMTP `mail.smtp.usetls`	Indicate if STARTTLS is needed for the mail server. Default value is "false". You should change it to "true" if your SMTP requires STARTTLS
SMTP username `mail.smtp.username`	Type the username that will be used if you set the authentication for SMTP parameter to "true".
SMTP password `mail.smtp.password`	Type the password that will be used if you set the authentication for SMTP parameter to "true".
Sender address mail `mail.from`	Email address that will displayed as the sender's address.

If you have complex mail server configurations, you may want to check the Javamail API FAQ for more information.

Related content

- Recommended Configurations
- Transactions and Connections
- Purging Audit Logs (NXP_LOGS)
- Configuration Templates
- Logs Analysis
- Configuration Examples
- Setup
- Nuxeo Clustering Configuration
- Configuration Parameters Index (nuxeo.conf)

Configuration Examples

Here are some configuration use cases:

- Changing the Live Edit Default Version Incrementation
- Changing the Default Port (8080)

The use of the Admin Center is highlighted in the steps below. However, you can do the same configurations by editing the `nuxeo.conf` file manually.

Changing the Live Edit Default Version Incrementation

When users edit documents with Live Edit, the default behavior is that no version incrementation occurs. This default behavior can be changed and you can set what version number should be incremented when users save a document with Live Edit.

Configure default Live Edit version incrementation:

1. In the Admin Center, click on the **Setup** tab of system information section.
2. In the Advanced Settings, edit the value of the parameter "org.nuxeo.ecm.platform.liveedit.autoversioning" :
   - `minor` will instruct the server to automatically increment the minor version of the document,
   - `major` will instruct the server to automatically increment the major version of the document,
   - `none` will instruct the server to not increment the version of the document (this is the default value).
3. Click the button **Save**.

Changing the Default Port (8080)

Nuxeo applications run on the 8080 port by default. As it may be used by another application, you may need to change it.

Change the default port:

1. In the Admin Center, click on the **Setup** tab of system information section.
2. In the Advanced Settings, edit the value of the parameter "nuxeo.server.http.port".
3. Click the button **Save**.
4. Restart the server as indicated on top of the page.

Related pages in this documentation

- Configuration Templates
- Configuration Examples
- Recommended Configurations
- Setup
- Nuxeo Clustering Configuration
- Configuration Parameters Index (nuxeo.conf)

In other documentations

- LiveEdit makes MS Office slow to start (Nuxeo Technical Knowledge Base (FAQ))
- Setup Firefox protocol handler with LiveEdit 2 for MS Office and OpenOffice.org (Nuxeo Technical Knowledge Base (FAQ))
- LiveEdit icons are still available in Nuxeo after LiveEdit has been uninstalled (Nuxeo Technical Knowledge Base (FAQ))
- I can't view my websites and blogs (displays a message "The HTTP header field "Accept" with value...") (Nuxeo Technical Knowledge Base (FAQ))

Configuration Templates

Nuxeo applications integrate a configuration templates system to ease configuration and maintenance of configuration files.
Nuxeo comes with default templates which mainly provide database configurations, but the templates can be used for any configuration purpose.

Properly using that template system ensures your customization of Nuxeo exclusively resides in your nuxeo.conf, custom templates and plugin modules.
For instance, users can create templates for development, pre-production, and production environments; each template will include a different set of xml contributions (users, ldap integration, database used, ...).

Templates are located in the "templates" directory (`$NUXEO_HOME/templates`). To enable a configuration, such as database configuration, you just need to indicate which template to use in the Admin Center's Setup tab or in the nuxeo.conf configuration file.

Here are the templates provided by default:

- common: common template used by other templates;
- common-binding: (JBoss only), template used by other templates;

- common-deploydir: (JBoss only), template used by other templates;
- default: default Nuxeo configuration template for test purpose;
- https: (Tomcat only), not recommended template for making the server listen to port 443 (HTTPS);
- monitor: (JBoss only), activate the JBoss LogginMonitor service to log miscellaneous MBean informations;
- postgresql: PostgreSQL configuration template;
- postgresql-quartz-cluster
- mssql: MS SQL Server configuration template;
- mssql-quartz-cluster
- mysql: MySQL configuration template;
- oracle: Oracle configuration template;
- oracle-quartz-cluster
- custom: sample custom templates. Of course, this template is empty by default. One should copy it outside `$NUXEO_HOME` and adapt to his needs.

> For production environment, it is recommended to define your own custom template outside `$NUXEO_HOME`, as for `nuxeo.conf`. It must then be referenced in `nuxeo.conf` with its absolute path.

Technical Overview

A server is considered as already configured when it has a "config" directory.
When the "config" directory doesn't exist, templates will be used to generate all configuration files (config and datasources).

The template files contain defined parameters such as ${sample.parameter}.

Values for parameters replacement are calculated this way:

- If nuxeo.conf does not define `nuxeo.templates`, then `nuxeo.templates` equals "default" (the deprecated parameter `nuxeo.template` is still read for backward compatibility).
- The ${nuxeo.templates} value is used for determining the chosen template(s).
- For each value "nuxeo.template" of ${nuxeo.templates} (comma separated values, relative to "templates/" directory or absolute path), the corresponding file templates/${nuxeo.template}/nuxeo.defaults is read for defining new default values and maybe including other templates which are recursively parsed.
- The file templates/nuxeo.defaults is read for default values not already defined.
- The file nuxeo.conf is read for custom values (overwriting default values).

Configuration files are then generated by this way:

- For each comma separated value of nuxeo.templates and nuxeo.template.includes (let say sample.template), files in templates ${sample.template} are copied using the previously calculated values for replacing parameters.
- Every included template will potentially overwrite its precedents.

Related content:

- Adding Custom Templates
- Configuration Templates
- Connecting Nuxeo to the Database
- Configuration Parameters Index (nuxeo.conf)

Configuration Parameters Index (nuxeo.conf)

Here is a list of available parameters for `nuxeo.conf`. This list may not be exhaustive but it will be often updated.

Those parameters can be either environment parameters used by Nuxeo runtime or template parameters used for values replacement in configuration files.

Parameter	Default value ("|" separates possible values)	Description
`JAVA_HOME`	None. If undefined nuxeoctl script will try to discover it.	Path to Java home directory.

`JAVA_OPTS`	`-Xms512m -Xmx1024m -XX:MaxPermSize=512m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Dfile.encoding=UTF-8`	Optional values passed to the JVM. Nuxeo requires at least 1024 Mo in JVM heap size and 256Mo as maximum permanent size (512 recommended). Decreasing garbage collector frequency avoid having too much CPU usage (Sun Java specific options, recommended by JBoss).
`launcher.start.max.wait`	300	Since Nuxeo 5.4.1. Maximum time to wait for effective Nuxeo server start before giving up (applies on commands "start" and "restart").
`launcher.stop.max.wait`	60	Since Nuxeo 5.5. Maximum time to wait for effective Nuxeo server stop cleanly before using forced stop.
`launcher.override.java.tmpdir`	true	Since Nuxeo 5.4.1. Possible values: `true` or `false` If true, will set `java.io.tmpdir = nuxeo.tmp.dir`.
`nuxeo.log.dir`	log	Log directory (absolute or relative to `NUXEO_HOME`). Linux recommended path: `/var/log/nuxeo/...`
`nuxeo.pid.dir`	bin	Directory where to store Nuxeo PID file.
`nuxeo.data.dir`	data	Data directory (absolute or relative to NUXEO_HOME). It involves all data not being stored in database. Linux recommended path: `/var/lib/nuxeo/...`
`nuxeo.tmp.dir`	`server/default/tmp` (JBoss) `temp` (Tomcat) `tmp` (Jetty)	Location of the temporary files.
`nuxeo.force.generation`	`true` | `once`	If "`true`", will force generation of configuration files; otherwise they are only generated when not existing. If "`once`", will force one time and switch to false after successful generation. If "`false`", configuration changes are ignored.
`nuxeo.templates`	default	Comma separated list of templates to include. Templates paths are absolute or relative to `$NUXEO_HOME/templates/`. Available templates: postgresql, mysql, mssql, oracle, custom, ...

`nuxeo.bind.address`	0.0.0.0	Server binding address. "0.0.0.0" means "all available network interfaces". WARNING: when changing `nuxeo.bind.address`, you must accordingly change `nuxeo.loopback.url`.
`nuxeo.server.http.port`	8080	Server HTTP listen port.
`nuxeo.server.ajp.port`	8009	Server AJP listen port. This is not available on Jetty.
`nuxeo.server.jvmRoute`	nuxeo	Since Nuxeo 5.4.2. Server AJP route for load-balancing
`nuxeo.server.tomcat-admin.port`	8005	Deprecated since Nuxeo 5.6. Tomcat server's "admin" port. This is only useful if you have another tomcat server running and want to avoid port conflicts.
`nuxeo.server.tomcat_admin.port`	8005	Since Nuxeo 5.6. Tomcat server's "admin" port. This is only useful if you have another tomcat server running and want to avoid port conflicts.
`nuxeo.server.https.port`	8443	Server HTTPS listen port. This is only useful if you have modified the application server to use HTTPS.
`nuxeo.server.emptySessionPath`	false	(Tomcat only) Since Nuxeo 5.5, until 5.7.1. If set to true, all paths for session cookies will be set to /. May be useful to enable authentication on proxyfied WebEngine applications (see HTTP and HTTPS Reverse-Proxy Configuration). Removed since Nuxeo 5.7.2 (see http://tomcat.apache.org/migration-7.html#Session_cookie_configuration).
`org.nuxeo.ecm.instance.name`	Nuxeo 5.8	Server name.
`org.nuxeo.ecm.instance.description`	Nuxeo	Server description.
`org.nuxeo.ecm.product.name`	Nuxeo Platform	Product name, displayed in the page title on your browser.
`org.nuxeo.ecm.product.version`	5.8	

`org.nuxeo.dev`	false	Since Nuxeo 5.6, this property uses the "dev" mode when running the Nuxeo application. This parameter should not be set to `true` on a production server, as it disables some caches, and enables hot redeploy of some JARs (Studio JARs for instance). For more information about the dev mode, see How to do incremental deployment (hot reload) in the JSF-Seam layer. Before 5.6, setting this property to true stopped the runtime when an error occured at deployment. This behaviour has been removed from the dev mode and is now controlled by the property `org.nuxeo.runtime.strict`.
`org.nuxeo.prod`	false	Since Nuxeo 5.8, setting this property to "true" will display a quite visible warning message in the Admin Center, stating that this is a production instance. This is useful for administrators who are sometimes confusing their Nuxeo production server with their test server (not to rat anyone out).
`org.nuxeo.automation.trace`	false	Since Nuxeo 5.7.3, you can enable this mode if you'd like to display automation traces during runtime: • You'll benefit from exhaustive logs to debug all automation chain and/or operation execution. • The automation trace mode is disabled by default (not suitable for production). • It can be activated through JMX via `org.nuxeo:TracerFactory` MBean during runtime.
`org.nuxeo.automation.trace.printable`	*	Since Nuxeo 5.7.3, by default, all automation executions are 'printable' (appear in logs) when automation trace mode is on. • You can filter chain and/or operation execution trace printing by setting this property to chain name and/or operation separated by comma. • Comment this property to get all automation chains/operations back in printing (by default set to * (star))
`studio.snapshot.disablePkgValidation`	false	Since Nuxeo 5.7.1, this property makes it possible to disable the Studio snapshot package validation from the Admin Center (there was no validation before). Defaults to `false`.

`org.nuxeo.ecm.webapp.dashboard.mode`	auto	Defines the dashboard mode. There are three modes: • `auto`: (default value) let Nuxeo choose dashboard based on user browser capabilities. • `old`: force usage of the 'old' JSF based dashboard for all users. • `opensocial`: force usage of the new OpenSocial based dashboard for all users.
`*templateName*.target`	`server/default/deploy/nuxeo.ear`	Directory where *templateName* files will be deployed.
`mailservice.user`	nobody	(JBoss only) User for e-mail authentication.
`mailservice.password`	password	(JBoss only) Password for e-mail authentication.
`mail.store.protocol` `mail.transport.protocol`	pop3 smtp	Server protocol parameters for e-mailing.
`mail.user`	nobody	User who will receive e-mail (unused in Nuxeo).
`mail.store.host`	localhost	e-Mail server.
`mail.store.user`	anonymous	
`mail.store.password`	password	
`mail.debug`	false	Enable debugging output from the JavaMail classes.
`nuxeo.notification.eMailSubjectPrefix`	[Nuxeo]	Subject prefix in Nuxeo notification e-mails.
`nuxeo.notification.eMailSigner`	The Nuxeo team	Since 5.7.2. Signer of the sent e-mail.
`mail.transport.host`	localhost	SMTP gateway server.
`mail.transport.port`	25 (without authentication) 587 (with authentication) 465 (SSL)	e-Mail server port.
`mail.transport.usetls`	false	Use TLS for the SMTP connection.
`mail.transport.auth`	true	
`mail.transport.user`	anonymous	
`mail.transport.password`	password	
`mail.from`	noreply@nuxeo.com	The e-mail address will be sent from.
`nuxeo.db.name`	nuxeo | NUXEO	Database name.
`nuxeo.db.user`	sa | nuxeo	Database username.
`nuxeo.db.password`	(empty value) | password	Database password.
`nuxeo.db.host`	localhost	Database host URL.

`nuxeo.db.port`		3700 (DB2) 5432 (PostgreSQL) 3306 (MySQL) 1521 (Oracle) 1433 (MSSQL)
`nuxeo.db.jdbc.url`	(database-dependent)	Database JDBC connection URL for Nuxeo datasources, for instance `jdbc:postgresql://${nuxeo.db.host}:${nuxeo.db.port}/${nuxeo.db.name}`.
`nuxeo.db.validationQuery`		Database validation query, a `SELECT` statement used to check connections before using them, usually `SELECT 1`. Using this has a noticeable speed impact but makes connections resilient to network or sever problems.
`nuxeo.db.min-pool-size`	5	Database minimum pool size for Nuxeo datasources.
`nuxeo.db.max-pool-size`	20 (JBoss) 100 (Tomcat)	Database maximum pool size for Nuxeo datasources.
`nuxeo.vcs.min-pool-size`	0	Database minimum pool size for Nuxeo repository (VCS).
`nuxeo.vcs.max-pool-size`	20	Database maximum pool size for Nuxeo repository (VCS).
`nuxeo.vcs.blocking-timeout-millis`	100	Since Nuxeo 5.8. Database maximum wait time to get a connection from the pool when all connections are in use, for Nuxeo repository (VCS).
`nuxeo.vcs.idle-timeout-minutes`	10	Since Nuxeo 5.8. Database timeout after which connections not in use are removed from the pool, for Nuxeo repository (VCS).
`nuxeo.vcs.fulltext.disabled`	false	Since Nuxeo 5.8. Whether full text indexing and querying should be completely disabled in the repository. See VCS Configuration for details
`nuxeo.vcs.noddl`	false	Since Nuxeo 5.8. Where DDL generation should be disabled in the repository. See VCS Configuration for details.
`nuxeo.vcs.idtype`	varchar	Since Nuxeo 5.8. The type of id column. See VCS Configuration for details.
`nuxeo.url`	http://localhost:8080/nuxeo	Application URL (without final slash), should be the public URL of your server (ie: http://www.yourdomain.com/....)

`nuxeo.loopback.url`	http://localhost:8080/nuxeo	Since Nuxeo 5.4.1. Nuxeo URL, for connections from Nuxeo to itself (theme banks default). The port should be the same as `nuxeo.server.http.port`. Since Nuxeo 5.5, if not explicitly configured, the loop back URL is generated from `nuxeo.bind.address`, `nuxeo.server.http.port` and `org.nuxeo.ecm.contextPath` values.
`org.nuxeo.ecm.contextPath`	/nuxeo	Application context path. Before 5.6, you also have to accordingly rename all occurrences of nuxeo.xml (for Tomcat)
`org.nuxeo.ecm.platform.transform.ooo.host.name`	127.0.0.1	DEPRECATED.
`org.nuxeo.ecm.platform.transform.ooo.host.port`	8100	DEPRECATED.
`org.nuxeo.ecm.platform.transform.ooo.version`	2.2.1	DEPRECATED.
`org.nuxeo.ecm.platform.transform.ooo.enableDaemon`	true	DEPRECATED.
`jod.connection.protocol`	SOCKET	OpenOffice Connection protocol, either PIPE or SOCKET.
`jod.max.tasks.per.process`	200	Maximum task per Office instance before restarting it.
`jod.task.execution.timeout`	120000	Will stop the task if it s not completed after timeout.
`jod.task.queue.timeout`	30000	Will stop looking for the next task in the queue after timeout.
`jod.office.home`		Home directory of OpenOffice or LibreOffice.
`jod.office.ports`	2003	Since Nuxeo 5.5. When running in SOCKET mode, comma-separated list of ports used for the office connection.
`jod.office.pipes`		Since Nuxeo 5.5. When running in PIPE mode, comma-separated list of pipe names used for the office connection.
`jod.jpipe.lib.path`		Path to Jpipe library. Only used when connecting to OO through PIPE.
`jod.template.profile.dir`		Path to custom OO template directory.
`opensocial.gadgets.host`	localhost	
`opensocial.gadgets.port`	8080	
`opensocial.proxy.proxySet`	false	DEPRECATED since Nuxeo 5.6

`opensocial.proxy.proxyHost`		DEPRECATED since Nuxeo 5.6 (use default `nuxeo.http.proxy.host`)
`opensocial.proxy.proxyPort`		DEPRECATED since Nuxeo 5.6 (use default `nuxeo.http.proxy.port`)
`opensocial.proxy.user`		DEPRECATED since Nuxeo 5.6 (use default `nuxeo.http.proxy.login`)
`opensocial.proxy.password`		DEPRECATED since Nuxeo 5.6 (use default `nuxeo.http.proxy.password`)
`repository.clustering.enabled`	false	Activate clustering mode.
`repository.clustering.delay`	1000	When clustering is activated, defines the delay during which invalidations don't need to be processed (expressed in milliseconds).
`repository.binary.store`		Defines the folder where binaries are stored. Useful when using clustering or to change the location of binaries to another location.
`nuxeo.templates.parsing.extensions`	xml,properties	Deprecated since Nuxeo 5.6. Files extensions being parsed for parameters replacement when copying templates.
`nuxeo.plaintext_parsing_extensions`	xml,properties	Since Nuxeo 5.6. Files extensions being parsed for parameters replacement when copying templates.
`org.nuxeo.ecm.jboss.configuration`	default	JBoss configuration to use ("`default`", "`minimal`", "`all`", ...) Pay attention to the fact that this won't apply to templates defining their own "*`template`*`.target`" value (for instance, "default" template sets `default.target=server/default/deploy` without being aware of `org.nuxeo.ecm.jboss.configuration` value).
`zip.entry.encoding`	ascii	Choose how to encode filename when exporting documents to zip in the worklist.
`org.nuxeo.ecm.platform.liveedit.autoversioning`	none,minor,major	see Configuration Examples
`nuxeo.wizard.done`	true or false depending on the package	If set to false, will start a setup wizard before starting Nuxeo.
`nuxeo.http.proxy.host`		HTTP proxy host.
`nuxeo.http.proxy.port`		HTTP proxy port.
`nuxeo.http.proxy.login`		HTTP proxy login.
`nuxeo.http.proxy.password`		HTTP proxy password.

`facelets.REFRESH_PERIOD`	-1	Indicates to the compiler the number of seconds to wait between subsequent checks for changes in modified JSF facelets in a running application. Useful for facelet debugging. To disable this compiler check use a value of -1 which is a recommended value for production deployments as compiler checks have an impact on application performance. Since version 5.6, the parameter `org.nuxeo.dev` should be used instead as it forces this parameter to value "2".
`nuxeo.db.transactiontimeout`	300	Database transaction timeout in seconds (available for server tomcat only - Since Nuxeo 5.5)
`server.status.key`		Since Nuxeo 5.5. Secure key for connecting to server status monitoring servlet. It is randomly generated if not set.
`session.timeout`	60	Since Nuxeo 5.5. Session timeout (see web.xml session-timeout).
`nuxeo.updatecenter.disabled`	false (unset)	Since Nuxeo 5.5, Disable the Update Center feature.
`theme.useOldLocalConfiguration`	false (unset)	Since Nuxeo 5.5, Use the old local configuration for theme, selecting a theme page instead of a flavor.
`org.nuxeo.big.file.size.limit`	5Mi (unset)	Since Nuxeo 5.4.1, redirect onto the big file download url if size exceed limit
`org.nuxeo.ecm.platform.ui.web.auth.NuxeoAuthenticationFilter.isLoginNotSynchronized`	false (unset)	Since Nuxeo 5.5, disable login synchronization
`nuxeo.wizard.packages.url`		Since Nuxeo 5.5, defines the base url used by the Setup Wizard to get the packages.xml file describing the available software packages options
`nuxeo.wizard.skippedpages`	null	Since Nuxeo 5.5, comma separated list of pages that should be skipped inside the wizard
`nuxeo.pageprovider.default-max-page-size`	100	Since Nuxeo 5.6 (5.5-HF06 and 5.4.2-HF20), defines the default maxPageSize to use in pageProvider when no value is defined in the pageProvider contribution. Value '0' means no limit.
`org.nuxeo.dnd.upload.timeout`	30000	Since Nuxeo 5.7.1 (5.6-HF08), maximum time for uploading a file via Drag & Drop to the server.

`org.nuxeo.dnd.exec.timeout`	30000	Since Nuxeo 5.7.1 (5.6-HF08), maximum time for executing import of files uploaded via Drag & Drop
`org.nuxeo.dnd.extendedmode.timeout`	2000	Since Nuxeo 5.7.1 (5.6-HF08), mouse over time before switching to extended mode UI (setting to -1 disables the extended mode)
`org.nuxeo.query.builder.ignored.chars`	!#$%&'()+,./\\\\:-@{|}`^~	Characters that are escaped in queries
`nuxeo.jsf.numberOfConversationsInSession`	4	Since 5.7.2, Parameter to control the number of conversation states that are saved in session. Each conversation holds a number of view states that is defined by `nuxeo.jsf.numberOfViewsInSession`
`nuxeo.jsf.numberOfViewsInSession`	4	Since 5.7.2 (5.6-HF20) Parameter to control the JSF init parameter `com.sun.faces.numberOfViewsInSession`
`nuxeo.jsf.numberOfLogicalViews`	4	Since 5.7.2 (5.6-HF20) Parameter to control the JSF init parameter `com.sun.faces.numberOfLogicalViews`
`nuxeo.jsf.enableDoubleClickShield`	true (unset)	Since 5.7.3. Enables a shield on forms to prevent users from submitting twice the same form (accidental double-click). Default value was false in 5.7.3 and is true since 5.8.
`nuxeo.jsf.useAjaxTabs`	false (unset)	Since 5.8. Enables ajaxified tabs on document views.
`nuxeo.vcs.use-nulls-last-on-desc`	true (unset)	Since 5.8. Asks the database to use "NULLS LAST" when sorting DESC. True by default to get the same result order between different databases. Also turning this option to false enable PostgreSQL and Oracle to use an index on the sorted column which can be huge performance improvement.

Related content

- Configuration Templates
- Configuration Examples
- Recommended Configurations
- Nuxeo Clustering Configuration

Database

Nuxeo applications store most of their data in a SQL database. Several databases are supported, but they must be configured to work correctly.

This takes two steps:

1. Configure the database:
    - PostgreSQL (8.4 to 9.3),
    - Oracle (10g R2 (10.2.0.5) and 11g),

  - MS SQL Server (2005 or 2008).
2. Connect Nuxeo to the database.

Configuring PostgreSQL

Nuxeo supports the following PostgreSQL versions:
PostgreSQL 8.4, 9.0, 9.1, 9.2 and 9.3.

We always recommend that you use the latest stable version, which is PostgreSQL 9.3 at the time of this writing.

PostgreSQL 8.4 is extremely old and really should not be used. PostgreSQL 9.0 and 9.1 are already dated and should be upgraded for better performance.

The database needs to be configured to work properly with Nuxeo. Some settings **must** be changed in order for Nuxeo to work. Other settings *should* be changed in order for Nuxeo to have good performance.

This FAQ will give you some hints to configure your database, but please refer to your DBA or the PostgreSQL documentation for more information (http://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server).

Most settings have to be changed in the `postgresql.conf` file. Some SQL commands may have to be executed directly at the PostgreSQL console (`psql`).

Mandatory Changes

Two-Phase Commit

Nuxeo 5.6 uses two-phase commit by default and needs to have the `max_prepared_transactions` settings updated. You can use the same value as max_connections.

Note that if you are using a single data source (`nuxeo.db.singleDataSource=jdbc/NuxeoDS` default since 5.7) this is not anymore required.

```
max_prepared_transactions = 100
```

This change is mandatory for PostgreSQL >= 8.4 since prepared transactions are disabled by default. If you don't change this option you will have `javax.transaction.HeuristicMixedException` exceptions.

On this page

Implicit Casts

Jena (used for relations and comments) and jBPM (used for workflows) assume some implicit value casting in the SQL they generate. However since PostgreSQL 8.3 the database is much stricter than PostgreSQL 8.2 with respect to value casting.

To make Nuxeo work with PostgreSQL >= 8.3, you must therefore execute the following commands in your PostgreSQL console when connected to the `template1` database, so that any database created afterward will come with the required CASTs (if your database is already created, execute the commands in your database as well):

```
CREATE FUNCTION pg_catalog.text(integer) RETURNS text STRICT IMMUTABLE LANGUAGE SQL
AS 'SELECT textin(int4out($1));';
CREATE CAST (integer AS text) WITH FUNCTION pg_catalog.text(integer) AS IMPLICIT;
COMMENT ON FUNCTION pg_catalog.text(integer) IS 'convert integer to text';

CREATE FUNCTION pg_catalog.text(bigint) RETURNS text STRICT IMMUTABLE LANGUAGE SQL
AS 'SELECT textin(int8out($1));';
CREATE CAST (bigint AS text) WITH FUNCTION pg_catalog.text(bigint) AS IMPLICIT;
COMMENT ON FUNCTION pg_catalog.text(bigint) IS 'convert bigint to text';
```

This change is mandatory for PostgreSQL >= 8.3 since casts have been simplified. If you don't change this option you will have `operator does not exist` exceptions.

Possible errors if you don't update the casts as described above are:

```
org.postgresql.util.PSQLException: ERROR: operator does not exist: integer =
character varying

org.postgresql.util.PSQLException: ERROR: operator does not exist: bigint =
character varying

com.hp.hpl.jena.db.RDFRDBException: Exception while checking db format -
com.hp.hpl.jena.db.RDFRDBException: Internal SQL error in driver -
org.postgresql.util.PSQLException: ERROR: current transaction is aborted, commands
ignored until end of transaction block

com.hp.hpl.jena.db.RDFRDBException: Internal SQL error in driver -
org.postgresql.util.PSQLException: ERROR: current transaction is aborted, commands
ignored until end of transaction block
```

For further details, please see this url. You might also be interested in this migration helper.

Language plpgsql (PostgreSQL < 9.0)

If not already done, if you have PostgreSQL < 9.0 you must enable the `plpgsql` language:

```
CREATE LANGUAGE 'plpgsql';
```

Execute this on the `template1` database, so that any database created afterward will get the required language. If your database is already created, execute the command in your database as well.

If you get the following error then it just means that the language is already created (which is the case since PostgreSQL 9.0) and there is nothing further to do:

```
ERROR:  language "plpgsql" already exists
```

Create the Role and Database for Nuxeo

For instance (please change the password and the `nuxeo.conf` file of your instance accordingly):

```
$ createuser -U postgres -W -P nuxeo
$ createdb -U postgres -W -O nuxeo -E UTF8 nuxeo
```

Or from the psql command prompt:

```
CREATE ROLE nuxeo WITH PASSWORD 'nuxeo' LOGIN;
CREATE DATABASE nuxeo ENCODING 'UTF8' OWNER nuxeo;
```

Note that using the `UTF8` encoding for your database is important.

Performance Tuning

Shared Buffers and System Cache

One of the most important thing for PostgreSQL is to have lots of shared buffers along with free memory that can be used by the system cache.

Refer to the section "Adapt your configuration to your hardware" to get the correct value.

If you plan to use 1 GB of shared buffers, update the following property in your `postgresql.conf` file:

```
shared_buffers = 1GB
```

If you use PostgreSQL < 9.3 the shared memory must be available on the system side using sysctl, you need to enable a little bit more at the OS level, for instance try 1 GB + 128 MB:

```
sysctl kernel.shmmax=1207959552
```

Then restart the PostgreSQL.
If there is no enough shared memory you will have an explicit error message and you should try with a bigger kernel.shmmax value.

Once PostgreSQL is started the retained shmmax value, should be registered in the `/etc/sysctl.conf` file by adding the following line.

```
kernel.shmmax = <SHMMAX_VALUE>
```

PostgreSQL needs to know how much memory the system will use for disk caching. This is used as a hint when executing queries, this memory *is not allocated* by PostgreSQL.

To set `effective_cache_size` value, you need to run your application once and check how much memory is used by system cache. This can be done using the free command and using the `free` value for `-/+ buffers/cache`.

```
effective_cache_size = 1536MB
```

Memory for Workers

Increasing the `work_mem` parameter allows PostgreSQL to do larger in-memory sorts which is much faster than disk sorts. Have in mind that `work_mem` size will be taken by each connection (a pool of 20 connections will take up to 20 * work_mem).

```
work_mem = 12MB
```

Increasing the `maintenance_work_mem` will speed up the vacuum procedure.

```
maintenance_work_mem = 512MB
```

Buffering Writes

The default `wal_buffers` can be increase to improve write access time. Increasing the checkpoint segments and completion target helps to spread out the writes.

```
wal_buffers = 16MB
checkpoint_segments = 32
checkpoint_completion_target=0.8
```

Index vs Table Scan

The `random_page_cost` parameter influences this query planner's choice. The value to use depends on your disk IO, here are some advices:

```
# random_page_cost = 4 # Slow disk AWS EBS
# random_page_cost = 2 # Recent HD
# random_page_cost = 1 # SSD
```

Updating the Planner Statistics

PostgreSQL computes statistics on table content in order to plan for the best performance when executing queries with joins and complex filters. The default configuration in PostgreSQL <= 8.3 is `default_statistics_target` set to the value 10 which can lead to not accurate enough estimates. In 8.4 this value is now set to 100 by default. We recommend a higher value like 500:

```
default_statistics_target = 500
```

If the database is already populated you need to execute `ANALYZE` to update the statistics.

Vacuuming

The autovacuum is enabled by default since PostgreSQL 8.3.

Exceptionally, a full vacuum can be done at downtime to recover disk space, it should be followed with a `reindexdb` command.

Monitoring

We recommend the following setting to have a meaningful log in production

```
log_line_prefix = '%t [%p]: [%l-1] user=%u,db=%d '
log_min_duration_statement = 400
log_checkpoints=on
log_lock_waits=on
log_temp_files=0
log_autovacuum_min_duration=0
log_statement = ddl
track_functions=pl
```

Also to have an effective monitoring you should install the following extensions : pg_stat_statements, pg_buffercache

1. Install postgresql-contrib package.

```
sudo apt-get install postgresql-contrib
```

2. Login to you database as postgres user and create the extensions (require PostgreSQL >= 9.1).

```
sudo su postgres -c'psql -U postgres -d nuxeo -c"CREATE EXTENSION
pg_buffercache;"'
sudo su postgres -c'psql -U postgres -d nuxeo -c"CREATE EXTENSION
pg_stat_statements;"'
```

3. `Update the configuration.`

```
shared_preload_libraries = 'pg_stat_statements, auto_explain'
# custom_variable_classes = 'pg_stat_statements, auto_explain' # uncomment if
you are on PostgreSQL 9.1
pg_stat_statements.max = 10000
pg_stat_statements.track = top
auto_explain.log_min_duration = -1
auto_explain.log_analyze = 'false'
```

4. Restart the database.

```
sudo /etc/init.d/postgres restart
```

See the PostgreSQL section of the Monitoring and Maintenance page.

Adapt Your Configuration to Your Hardware

Here are some values that can be used as a starting point for a dedicated server depending on the amount of memory.

Amount of RAM	4g	8g	16g	32g
shared_buffers	1g	2g	4g	8g
effective_cache_size	1536m	4g	8g	16g
work_mem	12m	12m	16m	20m
maintenance_work_mem	512m	1g	1g	1g
max_connections	63	103	153	203

Specific Configuration

Accent-Insensitive Full-Text Search

If you want accent-insensitive fulltext search, you'll need to install the *unaccent* contribution, create a new text search configuration, and specify its use in Nuxeo.

Unaccent is described here http://www.postgresql.org/docs/9.0/static/unaccent.html.

Install it by running `unaccent.sql` script. For Ubuntu users, this file is located at `/usr/share/postgresql/9.0/contrib/unaccent.sql`

Connect to your database and run the following instructions:

```
CREATE TEXT SEARCH CONFIGURATION fr ( COPY = french );
ALTER TEXT SEARCH CONFIGURATION fr ALTER MAPPING FOR asciihword, asciiword,
hword_asciipart, hword, hword_part, word WITH unaccent, french_stem;
```

Then replace in your `default-repository-config.xml` file the `french` analyzer by the one you just defined (`fr` in this example).

Mass Import Specific Tuning

When doing mass import you can disable the fulltext trigger and fulltext index. They must be dropped after a successful login on a running Nuxeo DM because DDL SQL commands are executed on the first access.

```
ALTER TABLE fulltext DISABLE TRIGGER nx_trig_ft_update;
DROP INDEX IF EXISTS fulltext_fulltext_idx;
DROP INDEX IF EXISTS fulltext_fulltext_description_idx;
DROP INDEX IF EXISTS fulltext_fulltext_title_idx;
```

After the import you can update the fulltext column like this:

```
ALTER TABLE fulltext ENABLE TRIGGER nx_trig_ft_update;
-- Let the trigger update the fulltext column
UPDATE fulltext SET fulltext = ''::TSVECTOR WHERE length(fulltext) is NULL;
-- For Nuxeo up to 5.4
CREATE INDEX fulltext_fulltext_idx ON fulltext USING gin (fulltext);
-- For Nuxeo >= 5.5
CREATE INDEX fulltext_fulltext_title_idx ON fulltext USING gin
(nx_to_tsvector(fulltext_title::character varying));
CREATE INDEX fulltext_fulltext_description_idx ON fulltext USING gin
(nx_to_tsvector(fulltext_description::character varying));
CREATE INDEX fulltext_fulltext_idx ON fulltext USING gin
(nx_to_tsvector(fulltext::character varying));
```

Changing *temporary* the PostgreSQL configuration during the import provides performance benefits:

```
full_page_writes = off
fsync = off
synchronous_commit = off
```

Please refer to the PostgreSQL reference manual.

Using uuid idType

If you want to use the PostgreSQL `uuid` type instead of the default `varchar(36)` (this is the case when you set `nuxeo.vcs.idtype=uuid` in the`nuxeo.conf` file) you need to create a new operator to support `GIN` index on `uuid[]` type.

```
CREATE OPERATOR CLASS _uuid_ops DEFAULT
   FOR TYPE _uuid USING gin AS
   OPERATOR 1  &&(anyarray, anyarray),
   OPERATOR 2  @>(anyarray, anyarray),
   OPERATOR 3  <@(anyarray, anyarray),
   OPERATOR 4  =(anyarray, anyarray),
   FUNCTION 1  uuid_cmp(uuid, uuid),
   FUNCTION 2  ginarrayextract(anyarray, internal, internal),
   FUNCTION 3  ginqueryarrayextract(anyarray, internal, smallint, internal,
internal, internal, internal),
   FUNCTION 4  ginarrayconsistent(internal, smallint, anyarray, integer, internal,
internal, internal, internal),
   STORAGE uuid;
```

Possible error if you don't create the operator described above is:

```
ERROR: data type uuid[] has no default operator class for access method "gin"
```

Reporting Problems

If you have a database configuration problem and you want to fill a JIRA ticket, there are some information to report:

- The PostgreSQL server state: is it dedicated or shared, which OS, how many CPU, RAM, is it a virtual machine...
- How much memory is available on the database server (`free -m` output).
- Amount of Nuxeo documents and PostgreSQL configuration. Using the following commands:
  1. Login on your database with the postgres user.

     ```
     sudo su - postgres
     ```

  2. Get the Nuxeo SQL script to dump your configuration.

     ```
     wget --no-check-certificate
     https://gist.github.com/bdelbosc/5507796/raw/dump-nuxeo-postgres-config.s
     ql
     ```

  3. Execute the SQL script with psql against the Nuxeo DB (not the default database named postgres).

     ```
     psql nuxeo -f dump-nuxeo-postgres-config.sql
     ```

- Attach the output file located in `/tmp/pgconf.txt` into the JIRA ticket. An example of such a result file is here, so that you can check that yours has the correct format.
- If you are monitoring the slowest queries (See monitoring section) you can zip and attach the `postgresql` log file to the JIRA ticket.

Related content in this documentation

- Connecting Nuxeo to the Database
- Configuring MS SQL Server
- Configuring Oracle
- Configuring PostgreSQL

Related pages in other documentation

- Configure Nuxeo 5.3 with VCS and PostgreSQL (Nuxeo Technical Knowledge Base (FAQ))
- I can't delete my PostgreSQL database (Nuxeo Technical Knowledge Base (FAQ))
- PostgreSQL limitations (Nuxeo Technical Knowledge Base (FAQ))

Configuring Oracle

Nuxeo supports the following versions of Oracle:
Oracle 10g R2 (10.2.0.5) and Oracle 11g R2 (11.2.0.1)

Oracle Text (fulltext)

Oracle Text needs to be enabled in your database for fulltext indexing, please consult your Oracle documentation.

If you fail to install Oracle Text, you will get on startup the following error:

```
java.sql.SQLException: ORA-29833: indextype does not exist
```

In addition, if you want to configure specific lexers or word lists then check http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/cdatadic.htm for configuration parameters and syntax. Lexers and word lists are used by Nuxeo when configured in its `default-repository-config.xml` file.

Grant on `DBMS_CRYPTO`

Since Nuxeo 5.3.2, you need to grant `DBMS_CRYPTO` execution (replace `nuxeo` with the database user):

On this page

```
GRANT EXECUTE ON SYS.DBMS_CRYPTO TO nuxeo;
```

Note that for Oracle running on Amazon RDS, `DBMS_CRYPTO` is now directly accessible and you should simply do:

```
GRANT EXECUTE ON DBMS_CRYPTO TO nuxeo;
```

This is due to optimizations now enabled on Oracle that need hashing functions (MD5), available in this package.

Possible errors if you don't do this grant are:

```
java.sql.SQLException: ORA-06550: line 1, column 7: PLS-00905: object
NUXEO.NX_REBUILD_READ_ACLS is invalid
```

Create Hibernate Sequence

If the first startup fails with the following error in your logs,

```
ERROR [org.hibernate.util.JDBCExceptionReporter] ORA-02289: sequence
does not exist
```

you need to run this statement as your Oracle user

```
CREATE SEQUENCE HIBERNATE_SEQUENCE;
```

Grant on `V$SESSION` and `GV$SESSION`

If you use Nuxeo Clustering (`repository.clustering.enabled=true`), then you must make sure that your database user has access to the system views `V$SESSION` and `GV$SESSION` (replace `nuxeo` with the database user):

```
GRANT SELECT ON SYS.V_$SESSION TO nuxeo;
GRANT SELECT ON SYS.GV_$SESSION TO nuxeo;
```

You can check that this works as intended by doing, as the database user:

```
SELECT SID FROM V$SESSION WHERE SID = SYS_CONTEXT('USERENV', 'SID');
SELECT SID FROM GV$SESSION WHERE SID = SYS_CONTEXT('USERENV', 'SID');
```

(`V$SESSION` is a public synonym for `SYS.V_$SESSION`.)

Note: the view `GV$SESSION` is used in recent Nuxeo version instead of `V$SESSION` to allow working with Oracle RAC.

Possible errors if you don't do this grant are:

```
java.sql.SQLException: ORA-00942: table or view does not exist
```

Grant for `CREATE TABLE`

As Nuxeo creates tables in the database at first startup, you need to grant `CREATE TABLE` to your database user.

```
GRANT CREATE TABLE TO nuxeo;
```

Other Grants

The following more standard grants must also be executed :

```
GRANT CONNECT TO nuxeo;
GRANT RESOURCE TO nuxeo;
```

The following is sometimes needed, if you have several schemas:

```
GRANT SELECT ANY TABLE TO nuxeo;
```

Restricted Environment with No GRANT

Some DBA will provide a restricted schema where there is no GRANT on your database.

In that case, you'll have to run this command, whereas usually DBMS_LOB is granted to public.

```
GRANT EXECUTE ON DBMS_LOB ON nuxeo;
```

To resume, here are the list of all GRANTs needed:

```
- CONNECT
- RESOURCE
- SELECT ANY TABLE
- CREATE TABLE
- CREATE PROCEDURE
- CREATE SEQUENCE
- CREATE TRIGGER
- CREATE TYPE
- CREATE VIEW
- EXECUTE ON DBMS_LOB
- EXECUTE ON DBMS_CRYPTO
```

Character Set

Your database must be configured with `NLS_CHARACTERSET` set to `AL32UTF8`. If your database character set is not `AL32UTF8`, you may observe incorrect behavior including:

- error while trying to insert null values into `acl_user` table (`ORA-01400: cannot insert NULL into ("HUDSON"."ACLR_USER"."USER_ID")`)
- incorrect storage of accented or special characters,
- no tree structure visible on the left of Nuxeo DM,
- queries returning no document.

To check the character set on your server, execute:

```
SELECT value FROM NLS_DATABASE_PARAMETERS WHERE parameter = 'NLS_CHARACTERSET';
```

If you need to change the character set of you database, please check http://download.oracle.com/docs/cd/B19306_01/server.102/b14225/ch11charsetmig.htm.

If for some reason you must use an unsupported character set that is not in the list: `AL32UTF8`, `UTF8`, `US7ASCII`, `WE8DEC`, `WE8ISO8859P1`, `WE8MSWIN1252`, then you will need an additional `orai18n.jar` JAR in your Java class path. Download `orai18n.jar` at http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html.
Then add it in the class path for your Nuxeo server. For instance, in JBoss, you just put the jar in `$JBOSS/server/default/lib`. (The file `orai18n.jar` replaces the `nls_charset*.*` files in the Oracle 9i and older releases.)

Technical explanation
Internally, for security checks, Nuxeo executes SQL statements that need to be passed ARRAY objects (for the list of principals and permissions), but if the correct character set is not installed then the Oracle JDBC driver behaves incorrectly and Oracle actually receives empty strings. This in turn results in empty results for the queries as none of the documents will match due to incorrect security checks. The orai18n.jar fixes this.

Import/Export

Starting 11gR2, Oracle does not allocate space for a table until the first row gets inserted into the table. What happens is if you take an export of the schema/database, the dump would not include any of the tables that hasn't got any space allocations yet. A configuration change needs to be done to allocate space even more tables with no records.

```
alter system set deferred_segment_creation=false;
```

JDBC Driver

Nuxeo needs the Oracle JDBC driver to establish a connection to the database.

The driver can be downloaded from the Oracle JDBC driver downloads site.
We recommand the latest version for 11.2.0.* : `ojdbc6.jar` for JDK 1.6. It is compliant with Oracle 10g.

The driver must be in the `$NUXEO_HOME/lib` directory.
If you are using the `oracle` template (`nuxeo.templates=oracle` in `nuxeo.conf`), just put the driver in the `$NUXEO_HOME/templates/oracle/lib` directory.

Reporting Problems

If you have a database configuration problem and you want to fill a JIRA ticket, there are some information to report:

- The Oracle server state: is it dedicated or shared, which OS, how many CPU, RAM, is it a virtual machine...
- How much memory is available on the database server (free -m output).
- Amount of Nuxeo documents and Oracle configuration. Using the "sqlplus" command line tool connect to your Nuxeo database and execute the following commands:

```
# Get the Nuxeo SQL script to dump your configuration
wget --no-check-certificate
https://gist.github.com/bdelbosc/7766893/raw/dump-nuxeo-oracle-conf.sql
# Run the script against your database
sqlplus nuxeo/nuxeo@NUXEO @dump-nuxeo-oracle-conf.sql
```

and attach the output file located in `/tmp/oraconf.txt` into the JIRA ticket.

Related content in this documentation

- Connecting Nuxeo to the Database
- Configuring MS SQL Server
- Configuring Oracle
- Configuring PostgreSQL

In other documentation

- I can't see tree structure with Nuxeo on Oracle!Queries returns no document on Oracle (Nuxeo Technical Knowledge Base (FAQ))
- Oracle limitations (Nuxeo Technical Knowledge Base (FAQ))

Configuring MS SQL Server

Nuxeo supports Microsoft SQL Server 2005, 2008, 2008 R2 and 2012 (including Azure).

Note that currently SQL Server 2008 may crash when working with full-text queries, see NXP-6143 for details. (This is a bug due to Microsoft, not Nuxeo.)

System Database Collation

We've observed incorrect behavior (in particular with full-text search) if the SQL Server `master` database is not configured with a case-insensitive collation (a collation name with "`CI`").

To make sure this is the case, use:

```
SELECT collation_name FROM sys.databases WHERE name = 'master'
```

For instance the following collations have been checked to work correctly:

- `SQL_Latin1_General_CP1_CI_AS`
- `French_CI_AS`

On this page

Database Collation

To work properly Nuxeo need to have some columns with a case-sensitive collation.

To make sure this is the case, use:

```
SELECT collation_name FROM sys.databases WHERE name = 'nuxeo' -- or your custom
database name
```

You need a case-sensitive collation (a collation name with "`CS`"), like `French_CS_AS`.

If this is not the case for your existing database you can change it like this:

```
ALTER DATABASE nuxeo COLLATE French_CS_AS
```

If you get database error related to rights issue, you can set it as a single user owner:

```
ALTER DATABASE nuxeo SET SINGLE_USER
WITH ROLLBACK IMMEDIATE
GO
ALTER DATABASE nuxeo COLLATE French_CS_AS
GO
ALTER DATABASE nuxeo SET MULTI_USER
```

Row Versioning-Based Transaction Isolation

To prevent locking and deadlocking problems you need to enable the row versioning-based isolation levels. With row versioning readers do not block other readers or writers accessing the same data. Similarly, the writers do not block readers. However, writers will block each other. Before each statement Nuxeo add a `` `SET TRANSACTION ISOLATION LEVEL READ COMMITTED;` `` so statement sees only data committed before the query began.

To enable the row versioning submit the following SQL commands:

```
ALTER DATABASE nuxeo SET ALLOW_SNAPSHOT_ISOLATION ON;
ALTER DATABASE nuxeo SET READ_COMMITTED_SNAPSHOT ON;
```

Note that there must be no other open connection in the database until `ALTER DATABASE` is complete, otherwise the last command will hang. You can work around this (when executing the command from SQL Server Management Studio for instance) by adding `WITH ROLLBACK IMMEDIATE`:

```
ALTER DATABASE nuxeo SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIATE;
```

If you don't execute the above commands, you will get the following error at Nuxeo startup:

```
Snapshot isolation transaction failed accessing database 'nuxeo' because snapshot
isolation is not allowed in this database. Use ALTER DATABASE to allow snapshot
isolation.
```

Recovery Model

A recovery model is a database property that controls how transactions are logged, whether the transaction log requires (and allows) backing up, and what kinds of restore operations are available. Three recovery models exist: simple, full, and bulk-logged.
(see more details here: http://msdn.microsoft.com/en-us/library/ms189275.aspx)

By default, recovery mode is full, so you can get performance issues when enabling this mode

Mode View:

```
SELECT name, recovery_model_desc
   FROM sys.databases
      WHERE name = 'model' ;
GO
```

Mode Update (if FULL):

```
USE master ;
ALTER DATABASE model SET RECOVERY SIMPLE ;
```

Full-text

If you configure a full-text index in Nuxeo (which is the default), you will need to make sure that your SQL Server instance has full-text search configured (it's an optional component during installation). See http://msdn.microsoft.com/en-us/library/ms142571.aspx for details.

Failing to do this will provoke errors like:

SQL Server Msg 7601

```
Cannot use a CONTAINS or FREETEXT predicate on table or indexed view 'fulltext'
because it is not full-text indexed.
```

SQL Server Msg 7616

```
Full-Text Search is not enabled for the current database. Use sp_fulltext_database
to enable full-text search for the database. The functionality to disable and enable
full-text search for a database is deprecated. Please change your application.
```

The French version of these messages, for reference:

SQL Server Msg 7601

```
Impossible d'utiliser le prédicat CONTAINS ou FREETEXT sur table ou vue indexée
'fulltext', car il n'y a pas d'index de texte intégral.
```

SQL Server Msg 7616

```
La recherche en texte intégral n'est pas activée dans la base de données en cours.
Utilisez sp_fulltext_database pour l'activer sur cette base de données. La
fonctionnalité de désactivation et d'activation d'une recherche en texte intégral
pour une base de données est désapprouvée. Modifiez votre application.
```

You can verify if your MSSQL instance has its full-text feature installed before creating your database:

```
SELECT SERVERPROPERTY('IsFullTextInstalled');
```

Full-Text Catalog

Nuxeo uses a full-text catalog named `nuxeo` by default, this can be changed in the Nuxeo configuration files (see configuration details).

Full-Text Analyzer

The language used to analyze full-text (called a LANGUAGE in SQL Server parlance) can be specified in the configuration for the database, instead of "english" in the section `<fulltext analyzer="english">`. The available languages in your database can be listed by using:

```
SELECT alias FROM sys.syslanguages
```

Additional Maintenance Operation

Since 5.4.2 HF05 the SQL Server back end comes with ACL (Access Control List) optimization. This optimization works with cache tables to store rights for each users and keep tracking of documents and rights changes. Theses data are reset when the server is started.

For long running instance or if you want to perform a hot backup without these unnecessary data, you can invoke the following stored procedure:

```
USE nuxeo;
EXEC dbo.nx_vacuum_read_acls;
```

Or you can exclude the following tables from your backup:

- `aclr`
- `aclr_modified`
- `aclr_permissions`
- `aclr_user_map`
- `aclr_user`

Deadlock and Lock Escalation

SQL Server is doing lock escalation: converting many row level locks to page lock or table lock. When doing concurrent write operations this can creates deadlock even when working on distinct data.

You can have more information on deadlock by enabling the following traces:

```
DBCC TRACEON(1222,-1);
 DBCC TRACEON(1204,-1);
```

Then you can try to disable the lock escalation on the table impacted by deadlocks:

```
ALTER TABLE mytable SET (LOCK_ESCALATION=DISABLE)
```

Clustered Index

SQL Server uses a clustered index to defined how the data is organized physically on disk. Before Nuxeo 5.7 we didn't define a clustered index, so the primary key is used, however this primary key is a random UUID which means that data keeps getting reorganized on disk on practically every insert or delete.

This has been fixed for new instance since Nuxeo 5.7. For instance created before there are migration script to apply to add these index, see NXP-10934 attachments to get the script.

Indexes Maintenance

If the indexes are fragmented then the query response will be slow and the data storage will require more disk space. Microsoft recommends reorganizing an index with a fragmentation between 5% and 30%, and rebuilding an index with a fragmentation of more than 30%. Database administrators should always make sure that fragmentation of indexes is handled on time.

Related Content in This Documentation

- Connecting Nuxeo to the Database
- Configuring MS SQL Server
- Configuring Oracle
- Configuring PostgreSQL

In Other Documentation

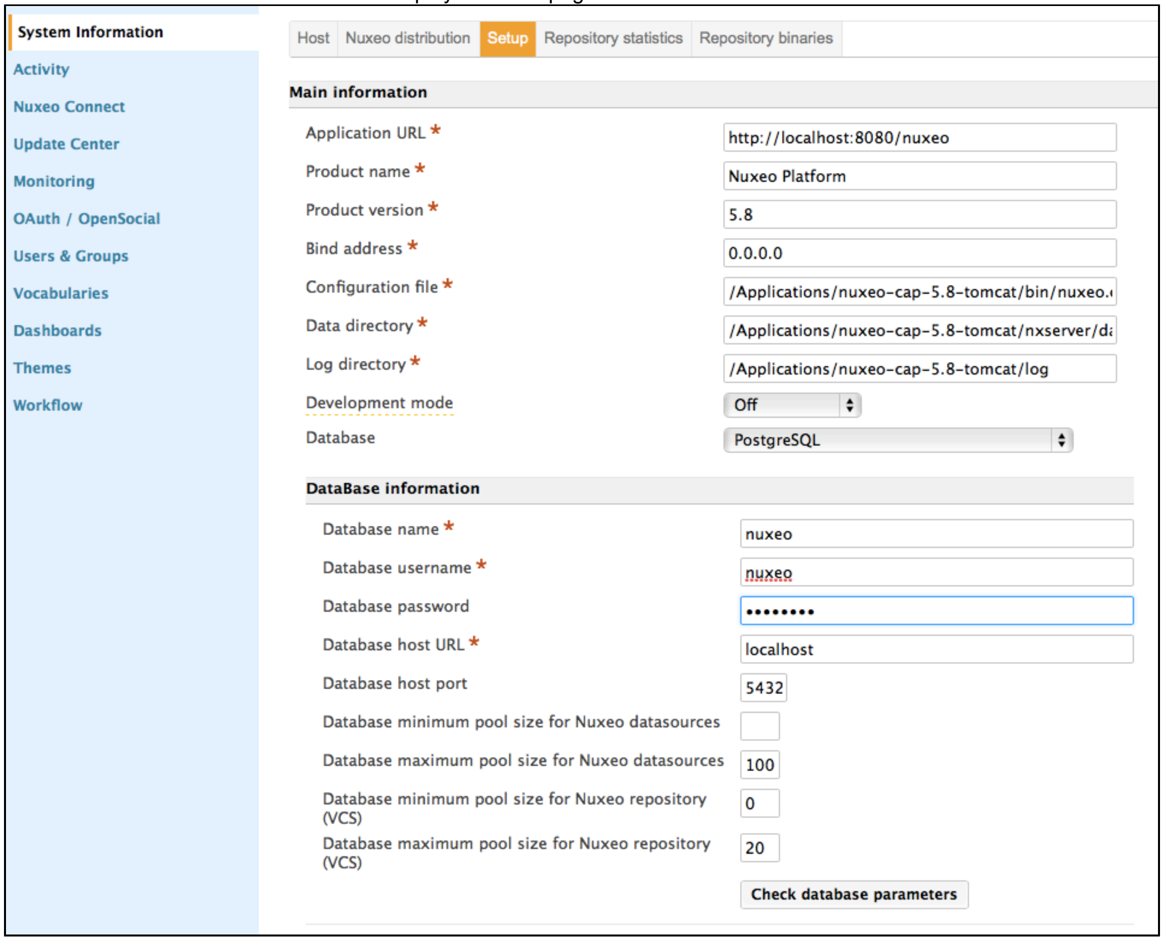
- SQL Server limitations (Nuxeo Technical Knowledge Base (FAQ))

Connecting Nuxeo to the Database

To connect Nuxeo to your database, you need to tell Nuxeo which database template to use and provide the database connection information.

Connecting Nuxeo to the Database From the Admin Center

1. In the Admin Center, click on the **Setup** tab of system information section.
2. In the **Main information** section, select the target database in the drop down menu.
   A new **Database Information** section is displayed on the page.



Main information

Application URL *	http://localhost:8080/nuxeo
Product name *	Nuxeo Platform
Product version *	5.8
Bind address *	0.0.0.0
Configuration file *	/Applications/nuxeo-cap-5.8-tomcat/bin/nuxeo.c
Data directory *	/Applications/nuxeo-cap-5.8-tomcat/nxserver/da
Log directory *	/Applications/nuxeo-cap-5.8-tomcat/log
Development mode	Off
Database	PostgreSQL

DataBase information

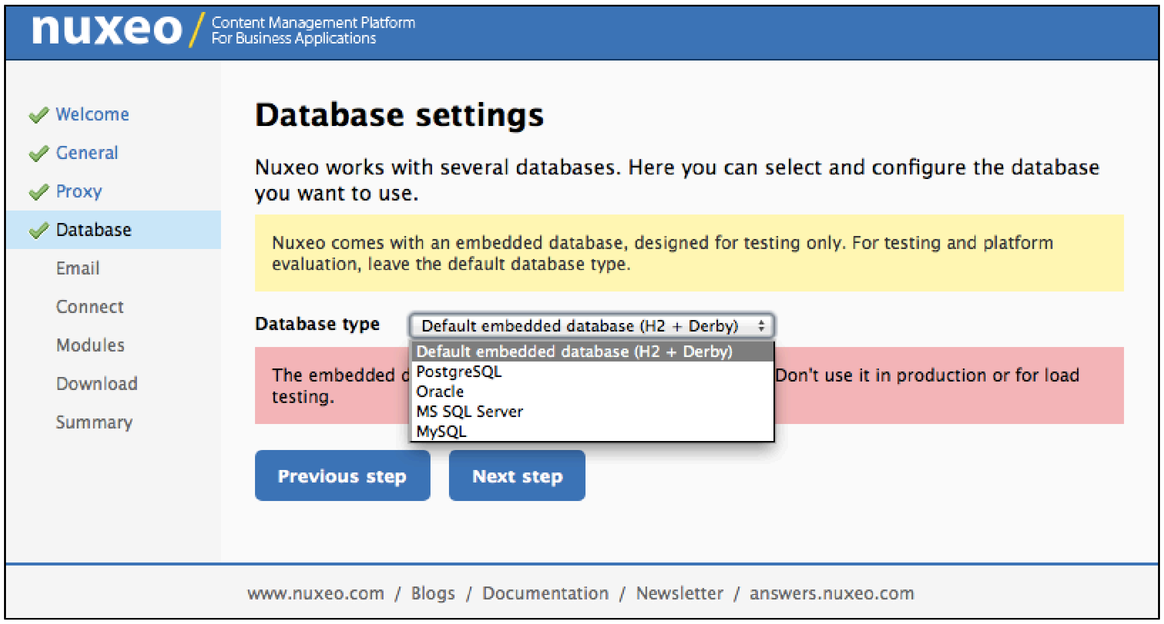
Database name *	nuxeo
Database username *	nuxeo
Database password	••••••••
Database host URL *	localhost
Database host port	5432
Database minimum pool size for Nuxeo datasources	
Database maximum pool size for Nuxeo datasources	100
Database minimum pool size for Nuxeo repository (VCS)	0
Database maximum pool size for Nuxeo repository (VCS)	20

3. Fill in the database connection information.
4. Click on the **Save** button.
5. Restart your server.

On this page

Connecting Nuxeo to the Database From the Startup Wizard

The first time you start your Nuxeo server, a wizard is displayed to help you setup your application. Step 3 is about the database: select the database you want to use in the drop down list and provide the connection information to the database.


nuxeo/ Content Management Platform For Business Applications
Welcome
General
Proxy
Database
Email
Connect
Modules
Download
Summary
Previous step
Next step
www.nuxeo.com / Blogs / Documentation / Newsletter / answers.nuxeo.com

Database settings

Nuxeo works with several databases. Here you can select and configure the database you want to use.

Nuxeo comes with an embedded database, designed for testing only. For testing and platform evaluation, leave the default database type.

Database type: Default embedded database (H2 + Derby)

Default embedded database (H2 + Derby)
PostgreSQL
Oracle
MS SQL Server
MySQL

The embedded d... Don't use it in production or for load testing.

Connecting Nuxeo to the Database From the `nuxeo.conf` File

By default, the "default" template is enabled on your Nuxeo server (see the Database templates section for more information on this template). You need to edit it to change the template to be used.

1. Open your `nuxeo.conf` file with a text editor.

   For Windows users
   Do not use Office writers, nor Notepad.
   Wordpad is fine, Notepad++ and SciTE are good text editors, there are a lot of other text editors.

2. If needed, uncomment or edit the `nuxeo.templates` parameter and replace `default` with the wanted database template's name.
3. Uncomment or edit the parameters below and provide their values:
   - `nuxeo.db.name`
   - `nuxeo.db.user`
   - `nuxeo.db.password`
   - `nuxeo.db.host`
   - `nuxeo.db.port`

   For production or load testing environments
   These are the minimum required parameters to enable the Nuxeo server to communicate with the database. For a production or load testing environment, you may need to provide the other commented parameters.

4. Save your modifications.
5. Restart the server.

Database Templates

The default available database templates are:

- default
- postgresql (recommended)
- oracle
- mssql
- mysql

default

This is the default Nuxeo configuration. It is designed for development or test purpose.

Repository backend: H2
Services backend: Derby

postgresql (recommended)

This is the recommended configuration for production, based on PostgreSQL.

Repository backend: PostgreSQL XA
Services backend: PostgreSQL XA

The PostgreSQL driver is included in the Nuxeo applications by default. However, if needed you can download it from http://jdbc.postgresql.org/download.html#current.
The JAR (for instance `postgresql-9.0-801.jdbc4.jar`) is located in `$JBOSS/server/default/lib/` or `$TOMCAT/lib/`.

> ✓ You can use a later driver with an earlier database version, for instance the 9.0 driver still works with PostgreSQL 8.3 or 8.4.

See the page Configuring PostgreSQL for more information on the database configuration.

oracle

Repository backend: Oracle XA
Services backend: Oracle

The driver is not included in Nuxeo applications. To install it:

1. Download the appropriate JDBC driver from: http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.
2. The JAR must be placed in `$JBOSS/server/default/lib/` or `$TOMCAT/lib/`.

> ✓ Nuxeo applications have been tested with the JDBC drivers available at http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html.

See the page Configuring Oracle for more information on the database configuration.

mssql

Repository backend: Ms SQL Server XA
Services backend: Ms SQL Server XA

The Open Source JTDS driver must be used (the official Microsoft JDBC driver has problems).
You can download it from: http://repo2.maven.org/maven2/net/sourceforge/jtds/jtds/1.2.2/jtds-1.2.2.jar.

This JAR must then be placed in `$JBOSS/server/default/lib/` or `$TOMCAT/lib/`.

See the page Configuring MS SQL Server for more information on the database configuration.

mysql

Repository backend: MySQL XA
Services backend: MySQL

The JDBC driver (downloadable from http://www.mysql.com/downloads/connector/j/) is included in Nuxeo applications and is located in `$TOMCAT/lib/`.

We recommend using a near-infinite wait_timeout in the MySQL server configuration.

Related content in admin documentation

- Connecting Nuxeo to the Database
- Adding Custom Templates
- Configuration Templates
- Configuring MS SQL Server
- Configuring Oracle
- Configuring PostgreSQL
- Configuration Parameters Index (nuxeo.conf)

Related topics in other documentation

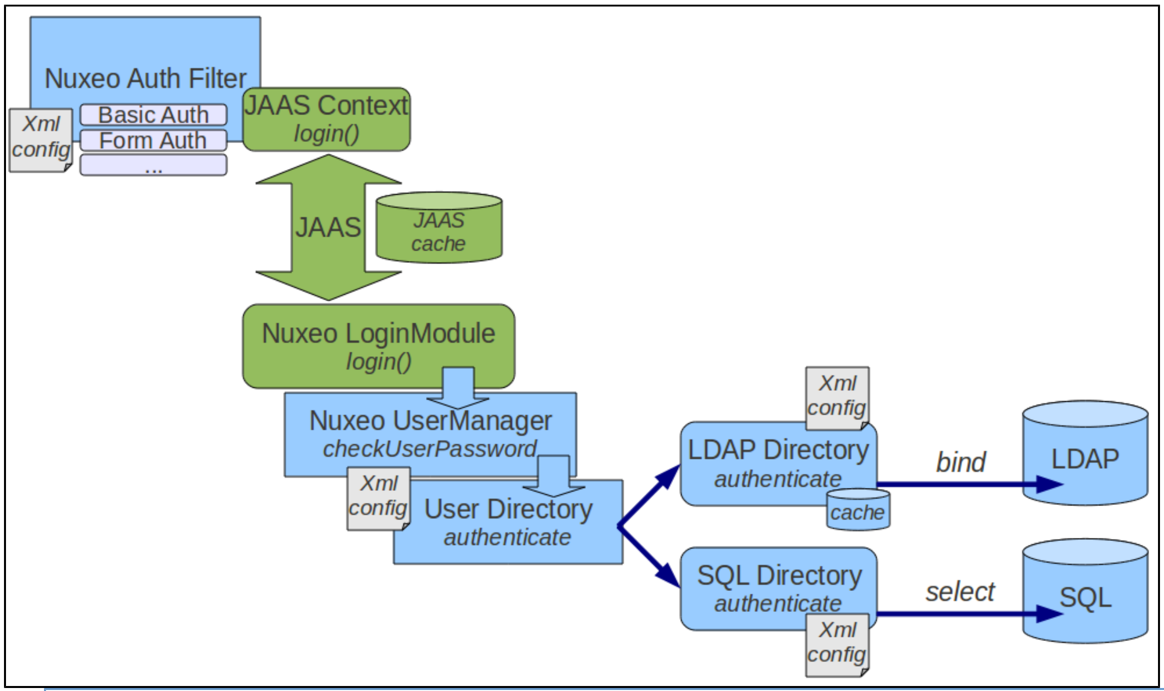
- Database limitations (Nuxeo Technical Knowledge Base (FAQ))
- SQL Server limitations (Nuxeo Technical Knowledge Base (FAQ))
- PostgreSQL limitations (Nuxeo Technical Knowledge Base (FAQ))
- H2 limitations (Nuxeo Technical Knowledge Base (FAQ))
- Oracle limitations (Nuxeo Technical Knowledge Base (FAQ))
- MySQL limitations (Nuxeo Technical Knowledge Base (FAQ))

Authentication, users and groups

Authentication and the Nuxeo Platform

The Nuxeo Platform authentication infrastructure is based on the JAAS standard and has be designed as pluggable as possible so that you can choose you you retrieve user information (identification) and how you validate (authentication).

You can see below a schema showing how the global auth process works:

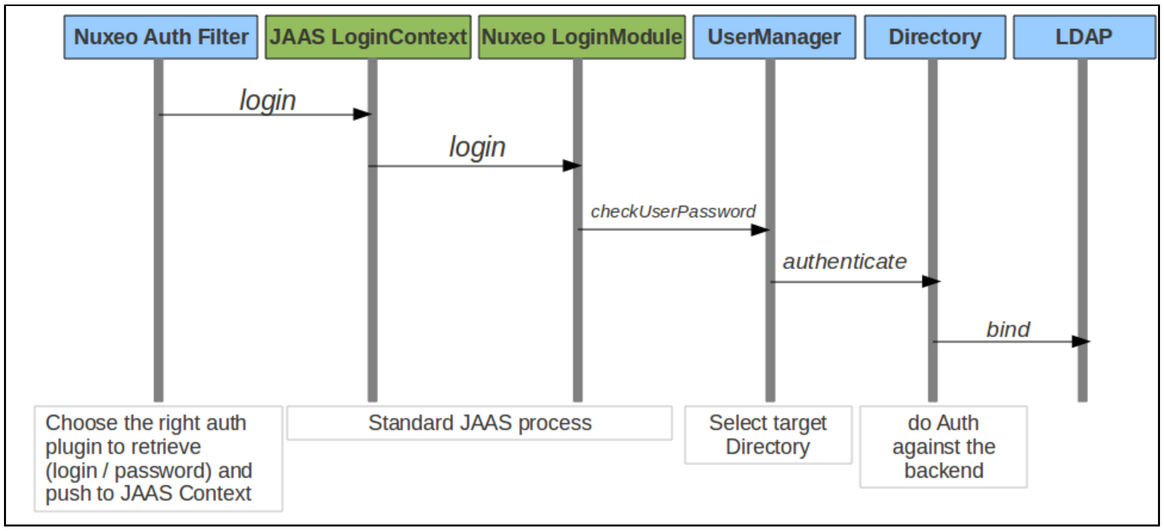


In this section

The blue blocks represents the pluggability points:

- retrieving user related information (getting login/password, getting a SSO ticket ...),
- validating user credentials against a backend (SQL DB, LDAP directory, external application ...).

You can see below the flow chart for an authentication.



Built-in Login Plugins

Login plugins are responsible for retrieving the user informations. It's usually a negotiation between the Nuxeo server and the client browser, but a SSO server may also be part of the process.

By default Nuxeo includes three login plugins:

- HTTP Basic authentication,
- Form based authentication,
- Anonymous authentication.

Additional login plugins are available as addons.

When needed, the security filter will determine the right login plugin to use according to:

- what the client browser can provide,
- the server configuration (that can be server wide or specific for some URLs).

Built-in `LoginModule` Plugins

The Nuxeo Platform uses extension points to define `LoginModule` plugins in order to avoid having to define several `LoginModules`.

By default there are two implementations of the `LoginModule` plugins:

- one that checks Login/Password against the declared directories (SQL oer LDAP),
- one that does not check the password and only checks that user exists and fetch user properties.
  This one is useful when Nuxeo is behind a portal or a SSO server and Nuxeo can not check any password.

OAuth support

Since version 5.4.1, the Nuxeo Platform provides a built-in support for OAuth.
Please see the dedicated OAuth Page for more info.

Available authentication modules

nuxeo-platform-login-cas2

> *The Central Authentication Service (CAS) is a single sign-on protocol for the web. Its purpose is to permit a user to access multiple applications while providing their credentials (such as user ID and password) only once. It also allows web applications to authenticate users without gaining access to a user's security credentials, such as a password. The name CAS also refers to a software package that implements this protocol.*
>
> (extracted from Wikipedia)

The `nuxeo-platform-login-cas2` defines an authentication plugin to validate the identity using the CAS server.
For further information, see Using CAS2 Authentication.

`nuxeo-platform-login-mod_sso`

This plugin is used when Nuxeo is behind a reverse proxy that manages the authentication and simply transmits user information as a set of

HTTP headers to Nuxeo.

This is typically the case when:

- Client Certificate authentication is used (Apache does the certificate validation and only transmit a DN to Nuxeo),
- a custom proxy-SSO is used.

`nuxeo-platform-login-kerberos`

This plugin provides SPNEGO/Kerberos authentication for Nuxeo. Please read this documentation to get started with this plugin.

`nuxeo-platform-login-ntlm`

This plugin allows NTLM V1 challenge/response over HTTP.

This plugin does not support NTLM V2 over HTTP and for recent MS Windows auth integration, you should probably use `nuxeo-platform-login-kerberos`.

`nuxeo-platform-login-portal-sso`

This plugin is used when the Nuxeo Platform is accessed via an external app (like a portal) that wants to access Nuxeo data in the name of a given user.

Because in most cases the external app does not know the password of the user, this plugin allow to define a shared secret between the app and the Nuxeo Platform so that the app can access Nuxeo as if it was a given user.

`nuxeo-platform-login-shibboleth`

The Shibboleth® System is a standards based, open source software package for web single sign-on across or within organizational boundaries. It allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner.

The `nuxeo-platform-login-shibboleth` bundle defines:

- an authentication plugin to map the user metadata from HTTP headers,
- a `NuxeoExceptionHandler` to force the login of an anonymous user trying to access a restricted resource,
- ShibbGroups, virtual groups based on Shibboleth attributes manageable from the UI,
- a hierarchical group suggestion widget for the access rights management tab.

For further information, see Using Shibboleth.

Using a LDAP Directory

In Nuxeo, users and groups are managed by *directories*. If you want your Nuxeo instance to use a LDAP directory you will need to:

- configure a user directory pointing to your LDAP server(s),
- configure a group directory pointing to your LDAP server(s) (if you need LDAP groups).

Of course you can have a specific custom config where:

- you use a custom user / group schema,
- you use several LDAP directories, or a mix of SQL and LDAP directories.

But for the most common use case, all you want to do is map the default `userDirectory` to your LDAP Server. Since groups are used in Nuxeo to associate permissions with content, fetching groups from LDAP is usually not very efficient: LDAP groups are usually not designed for that.

On this page

Simple Configuration Example

1. Create a file called `default-ldap-users-directory-config.xml` in your config directory:
    - `server/default/deploy/nuxeo.ear/config/` in JBoss,
    - `nxserver/config/` in Tomcat.
2. Then copy this content (make sure it's valid XML, sometimes what you think is a space character is actually a non-breaking space (`U+00`

A0) which is invalid in XML):

```
<?xml version="1.0"?>
<component name="org.nuxeo.ecm.directory.ldap.storage.users">

  <require>org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory</require>

  <!-- the groups SQL directories are required to make this bundle work -->
  <require>org.nuxeo.ecm.directory.sql.storage</require>

  <extension target="org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory"
    point="servers">

    <!-- Configuration of a server connection
      A single server declaration can point to a cluster of replicated
      servers (using OpenLDAP's slapd + sluprd for instance). To leverage
      such a cluster and improve availability, please provide one
      <ldapUrl/> tag for each replica of the cluster.
    -->
    <server name="default">
      <ldapUrl>ldap://localhost:389</ldapUrl>
      <!-- Optional servers from the same cluster for failover
        and load balancing:

        <ldapUrl>ldap://server2:389</ldapUrl>
        <ldapUrl>ldaps://server3:389</ldapUrl>

        "ldaps" means TLS/SSL connection.
      -->

      <!-- Credentials used by Nuxeo5 to browse the directory, create
        and modify entries.

        Only the authentication of users (bind) use the credentials entered
        through the login form if any.
      -->
      <bindDn>cn=nuxeo5,ou=applications,dc=example,dc=com</bindDn>
      <bindPassword>changeme</bindPassword>
    </server>
  </extension>

  <extension target="org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory"
    point="directories">
    <directory name="userDirectory">
      <server>default</server>
      <schema>user</schema>
      <idField>username</idField>
      <passwordField>password</passwordField>

      <searchBaseDn>ou=people,dc=example,dc=com</searchBaseDn>
      <searchClass>person</searchClass>
      <!-- To additionally restricte entries you can add an
        arbitrary search filter such as the following:


<searchFilter>(&amp;(sn=toto*)(myCustomAttribute=somevalue))</searchFilter>

        Beware that "&" writes "&amp;" in XML.
      -->
      <!-- use subtree if the people branch is nested -->
      <searchScope>onelevel</searchScope>

      <!-- using 'subany', search will match *toto*. use 'subfinal' to
        match *toto and 'subinitial' to match toto*. subinitial is the
        default  behaviour-->
      <substringMatchType>subany</substringMatchType>

      <readOnly>false</readOnly>

      <!-- comment <cache* /> tags to disable the cache -->
      <!-- cache timeout in seconds -->
      <cacheTimeout>3600</cacheTimeout>

      <!-- maximum number of cached entries before global invalidation -->
      <cacheMaxSize>1000</cacheMaxSize>

      <!--
           If the id field is not returned by the search, we set it with the
searched entry, probably the login.
           Before setting it, you can change its case. Accepted values are
'lower' and 'upper',
           anything else will not change the case.
      -->
      <missingIdFieldCase>lower</missingIdFieldCase>

      <!-- Maximum number of entries returned by the search -->
      <querySizeLimit>200</querySizeLimit>

      <!-- Time to wait for a search to finish. 0 to wait indefinitely -->
      <queryTimeLimit>0</queryTimeLimit>

      <creationBaseDn>ou=people,dc=example,dc=com</creationBaseDn>
      <creationClass>top</creationClass>
      <creationClass>person</creationClass>
      <creationClass>organizationalPerson</creationClass>
      <creationClass>inetOrgPerson</creationClass>

      <rdnAttribute>uid</rdnAttribute>
      <fieldMapping name="username">uid</fieldMapping>
      <fieldMapping name="password">userPassword</fieldMapping>
      <fieldMapping name="firstName">givenName</fieldMapping>
      <fieldMapping name="lastName">sn</fieldMapping>
      <fieldMapping name="company">o</fieldMapping>
      <fieldMapping name="email">mail</fieldMapping>

      <references>
        <inverseReference field="groups" directory="groupDirectory"
          dualReferenceField="members" />
      </references>
    </directory>
  </extension>

  <extension target="org.nuxeo.ecm.platform.usermanager.UserService"
point="userManager">
    <userManager>
      <defaultAdministratorId>johndoe</defaultAdministratorId>
      <defaultGroup>members</defaultGroup>
    </userManager>
```

```
  </extension>
</component>
```

3. Then you should edit this file:
   a. Set the correct server:
      - `<ldapUrl>`
      - `<bindDn>` and `<bindPassword>`
   b. Set the correct LDAP config:
      - `<searchBaseDN>`
      - `<searchClass>`
      - `<fieldMapping>`
   c. If you want Nuxeo to be able to create users in the LDAP directory:
      - make sure the user you use to access LDAP has write access,
      - define the `<creationBaseDn>` and associated parameters.
   d. Define the default mapping:
      - since the *Administrator* user won't exists anymore, you should assign at least one user to be administrator using `<defaultAdministratorId>`,
      - you can also choose to make all users members of the default "members" group using `<defaultGroup>`.
4. Restart the Nuxeo server, and you should now be able to authenticate against LDAP.

If you want to roll back the changes, simply delete the `default-ldap-users-directory-config.xml` file and restart the server.

For a more detailed view about possible configuration, see:

- LDAPDirectory and associated extension points,
- UserManager extension point.

The ldaptools/ folder in source code of the `nuxeo-platform-directory-ldap` module further provides sample LDIF files and OpenLDAP configuration file to help you setup a sample OpenLDAP server you can use as a base setup to build your corporate directory.

Using Active Directory

If you use Active Directory and want to use it with Nuxeo, you need to:

1. Be sure that LDAP mode is enabled on the Active Directory server,
2. Get the schema info (because Active Directory schema changes depending on a lot of external factors).

Once you have this information, you can connect Nuxeo to Active Directory as it was a real LDAP server.

Active Directory users are advised to use the aggregated global catalog port number (3268 by default) instead of the default LDAP port (389) in order to avoid getting referrals request to sub-directories blocked by corporate firewalls.

Usually with AD you will have to map the field "username" to "sAMAccountName".

Also, it happens that for the bindDN, it expects only an email adress, ex:

`<bindDn>applicative-account-nuxeo@toto.local</bindDn>`

Advanced Configuration

For more details on directories, such as configuring "multi-directories", see Directories and Vocabularies.

Known Issues

LDAP Contribution Not Activated

Since Nuxeo 5.4.2 and until version 5.5, a bug (fixed on version 5.6) prevents the LDAP contribution from being activated in some cases. See NXP-8852 and NXP-5574 for more information about the issue and the fix.

A quick workaround for this bug is to put in comments the `"<directory name="userDirectory">...</directory>"` part in `templates/common/config/default-sql-directories-bundle.xml` (or overwrite that file with a custom template).

A cleaner workaround is to define directories whose name are different from the default ones (userDirectory for users, groupDirectory for groups).
Then you need to use the user manager to specify the name of the directories which will be used for authentication, searching, ...
Therefore you should apply the changes described below to your existing LDAP contributions:

```
<!-- directory for users -->
<directory name="userLdapDirectory">
  (...)
  <inverseReference field="groups" directory="groupLdapDirectory"
          dualReferenceField="members" />
</directory>

<!-- directory for groups -->
<directory name="groupLdapDirectory">
    (...)
    <ldapReference field="members" directory="userLdapDirectory"
forceDnConsistencyCheck="false" staticAttributeId="uniqueMember"
dynamicAttributeId="memberURL"/>

    <ldapReference field="subGroups" directory="groupLdapDirectory"
forceDnConsistencyCheck="false" staticAttributeId="uniqueMember"
dynamicAttributeId="memberURL"/>
    (...)
</directory>

<!-- definition in the user manager -->
<extension target="org.nuxeo.ecm.platform.usermanager.UserService"
point="userManager">
  <userManager>
    (...)
    <users>
      <directory>userLdapDirectory</directory>
    </users>
    (...)
    <groups>
      <directory>groupLdapDirectory</directory>
    </groups>
    (...)
  </userManager>
</extension>
```

See attached files for templates of LDAP configuration.

This method applies to multi-directories too.

Debug Information

If you encounter some difficulties configuring LDAP, the first step is to get more details about what happens.

In the Log4J configuration, increase the log level for `org.nuxeo.ecm.directory` and `org.nuxeo.runtime.model.impl`:

```
<category name="org.nuxeo.ecm.directory">
  <priority value="DEBUG" />
</category>
<category name="org.nuxeo.runtime.model.impl">
  <priority value="INFO" />
</category>
```

This will give you more informations such as:

- Is your XML contribution properly loaded? Search for the component name of your contribution in the log file (for instance "org.nuxeo.ecm.directory.ldap.storage.users").
- Did the LDAP directory initialized? If so, your "servers" extension point is working.

- What is the LDAP request sending when you try to log in Nuxeo? You must be run the same request outside Nuxeo, using your preferred LDAP tool.

Apache Directory Studio can be used to replicate the LDAP requests sent by Nuxeo to the LDAP server and check their responses. If you seek help on answers.nuxeo.com or connect.nuxeo.com please include the LDIF export of a sample user entry and a sample group entry (if you want to use the LDAP server to resolve the groups).

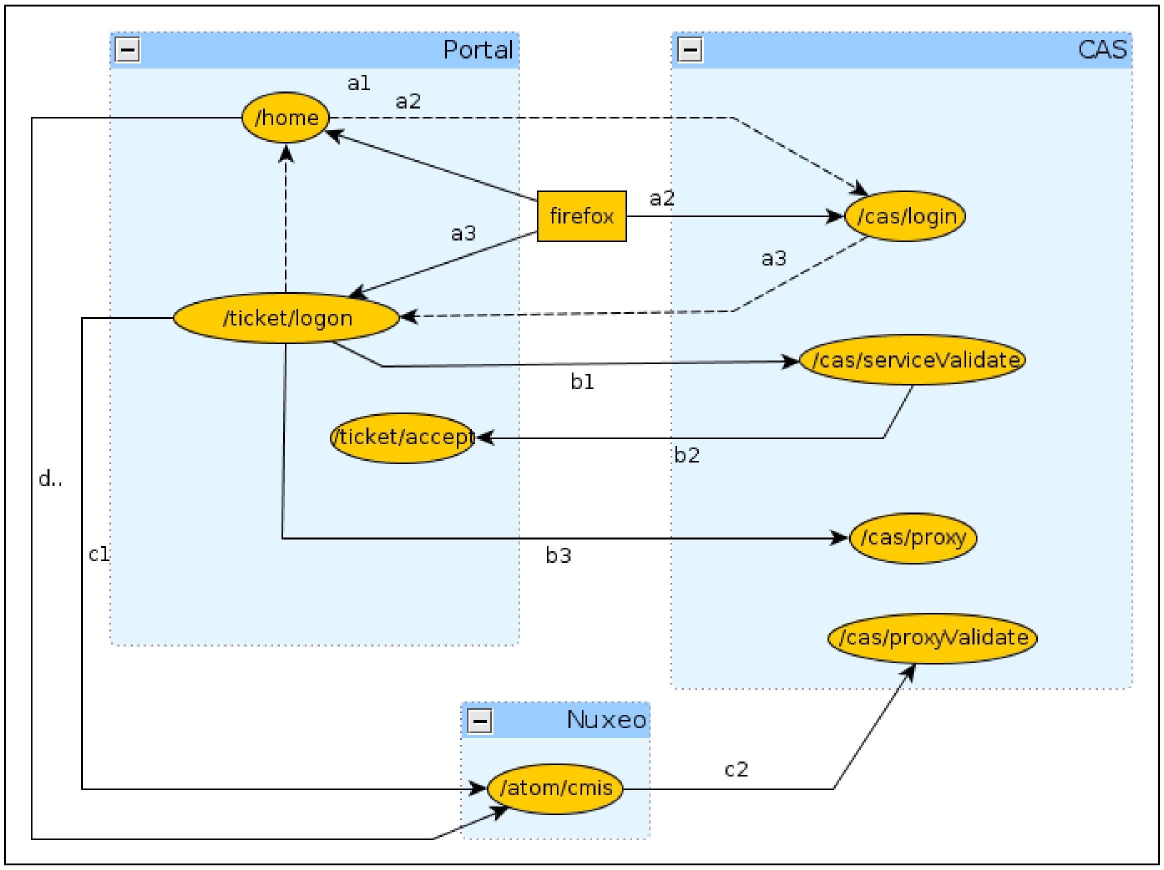
Related sections

Using a LDAP Directory (Nuxeo Installation and Administration - 5.8)
Adding Custom LDAP Fields to the UI (Nuxeo Platform Developer Documentation - 5.8)
Directories and Vocabularies (Nuxeo Platform Developer Documentation - 5.8)

Using CAS2 Authentication

A typical CAS use case would be the portal. In this n-tiers architecture, the identity is to be shared between the components.

The following diagram depicts the interactions between a client, a portal, a CAS server and Nuxeo for establishing the authentication context.



(a) Portal Service Ticket

The first phase is the portal authentication (a1).

```
GET /home
```

The client is redirected to the CAS server for entering its credentials (a2).

```
GET /cas/login?service=http://127.0.0.1:9090/ticket/validate
```

Once the credentials are entered, if they are valid, the CAS server generates a service ticket `ST`.
The client is redirected by the CAS server back onto the portal using the `service` URL (a3).

In the same time, the CAS server generates a ticket granting and registers it client side using the cookie `CASTGC`. If the cookie is already present in the request headers, then the client is automatically redirected to the portal.

```
http://127.0.0.1:9090/ticket/validate?ticket=ST-81-rCbbm5oj9geCKjvhNCvJ-
cas
```

(b) Proxy Granting Ticket

In the second phase, the portal validates the service ticket and requests for a proxy granting ticket `PGT` (b1).

```
GET /cas/serviceValidate?ticket=ST-81-rCbbm5oj9geCKjvhNCvJ-cas&

service=http://127.0.0.1:9090/ticket/validate&pgtUrl=http://127.0.0.1:90
90/ticket/accept
```

If the ticket is valid, the CAS server invokes the `pgtUrl` callback with two parameters `pgtIou` and `pgtId` (b2).

```
GET /ticket/accept?pgtIou=PGTIOU-34-jJZH23r2wbKUqbc3dLFt-cas&
    pgtId=TGT-45-sSnfcQ7A0TXGsQR2gJONm74rObZ0qRQzhENJWTdZJG5rcGN2T5-cas
```

In case of success, the server responds to the portal with the following content

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
.<cas:authenticationSuccess>
..<cas:user>slacoin</cas:user>
..<cas:proxyGrantingTicket>PGTIOU-34-jJZH23r2wbKUqbc3dLFt-cas</cas:proxyGrantingTick
et>
.</cas:authenticationSuccess>
</cas:serviceResponse
```

The `pgtIou` is used portal side for retrieving the accepted `PGT`.

(c) Nuxeo Proxy Ticket

In the third phase, the portal asks the CAS server for a new service ticket that will give him access to the Nuxeo server using the client identity (c1).

```
GET
/cas/proxy?pgt=TGT-45-sSnfcQ7A0TXGsQR2gJONm74rObZ0qRQzhENJWTdZJG5rcGN2T5
-cas&
    targetService=http://127.0.0.1:8080/nuxeo/atom/cmis
```

The CAS server generates a new ST and responds to the portal with the following content:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
.<cas:proxySuccess>
..<cas:proxyTicket>ST-82-20eCHgCqvMCvnP6AmZmz-cas</cas:proxyTicket>
.</cas:proxySuccess>
</cas:serviceResponse>
```

Then the proxy ticket is used by the portal for login into Nuxeo (c2).

```
GET /nuxeo/atom/cmis?ticket=ST-82-20eCHgCqvMCvnP6AmZmz-cas
    &proxy=http://127.0.0.1:9090/ticket/accept
    &service=http:127.0.0.1:8080/nuxeo/atom/cmis
```

The Nuxeo server validates the ticket by invoking the portal server (c3).

```
GET
/cas/proxyValidate?ticket=ST-82-20eCHgCqvMCvnP6AmZmz-cas&service=http:12
7.0.0.1:8080/nuxeo/atom/cmis
```

If the ticket is valid, the CAS server sends the following response:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
.<cas:authenticationSuccess>
..<cas:user>slacoin</cas>
..<cas:proxyGrantingTicket>PGTIOU-34-jJZH23r2wbKUqbc3dLFt-cas</cas:proxyGrantingTick
et>
..<cas:proxies>
...<cas:proxy>http://127.0.0.1:9090/ticket/accept</cas:proxy>
..</cas:proxies>
.</cas:authenticationSuccess>
</cas>
```

The Nuxeo server creates an HTTP session and sends the AtomPub response message.

```
<?xml version='1.0' encoding='UTF-8'?>
<app:service xmlns:app="http://www.w3.org/2007/app"
             xmlns:atom="http://www.w3.org/2005/Atom"
             xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
             xmlns:cmisra="http://docs.oasis-open.org/ns/cmis/restatom/200908/">
  <app:workspace>...</app:workspace>
</app:service>
```

The portal saves the client context for being able to invoke Nuxeo using the same HTTP session.

(d) Invoking Nuxeo

The Nuxeo HTTP session id is retrieved from the portal session context and invoked.

```
GET /nuxeo/atom/cmis?repositoryId=default
```

CAS2 and Anonymous Authentication

CAS2 and anonymous authenticators have flows that can interfere with each others, creating some side effects like bad redirections.

To avoid that, the CAS2 plugin provides a replacement for the default Anonymous authenticator : basically this is a *"CAS aware Anonymous authenticator"*. You can see a sample configuration available in: https://github.com/nuxeo/nuxeo-platform-login/blob/release-5.8/nuxeo-platform-login-cas2/Sample/CAS2-Anonymous-bundle.xml.

But, basically, wanting to put together both CAS2 and Anonymous authentication means you have two types of users that will access Nuxeo. So, an alternate approach is to define two separated authentication chains, one for each type of user:

- One chain for authenticated users: using CAS2 and some other authentication method you may need;
- One chain for anonymous access.

In most of the case, each type of user will have access via a separated virtual host at reverse proxy level. You can use this so that:

- At reverse proxy level you add a header for tagging the type of request;
  ex: Anonymous requests will have the header `X-anonymous-access` set to "on";
- At nuxeo level you configure the chains depending on the header;
  - Default / main chain is the one using CAS2;
  - You define specific chain for requests having the ``X-anonymous-access`.

Sample Xml contribution

```
<specificAuthenticationChain name="anonymous-access">
        <headers>
            <header name="X-anonymous-access">on</header>
        </headers>
        <allowedPlugins>
            <plugin>ANONYMOUS_AUTH</plugin>
        </allowedPlugins>
    </specificAuthenticationChain>
```

You can see specificChains extension point for more info.

Using OAuth

Background Information About OAuth

This section proposes a quick introduction to OAuth concepts.

For a detailed presentation of OAuth, you should read the OAuth 1.0 Protocol specification.

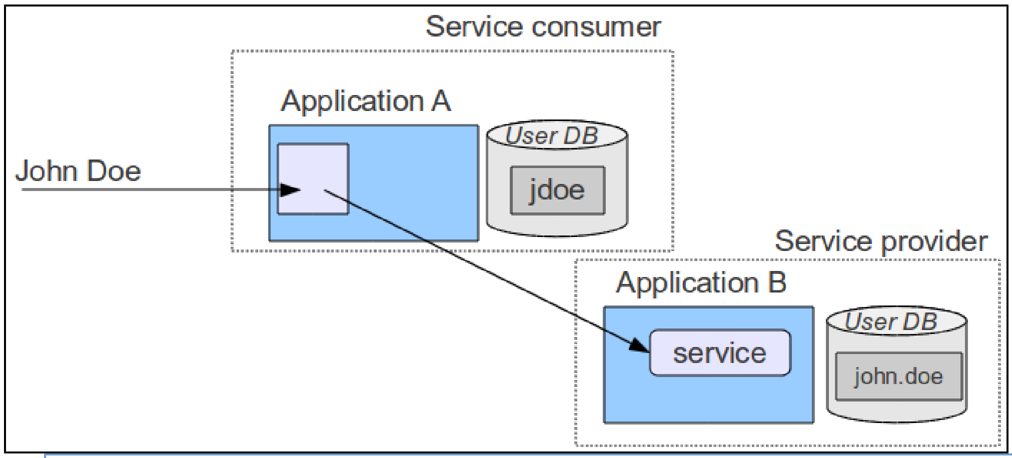
Problem to Solve

OAuth addresses authentication issues in cases that include:

- A Service Provider that exposes a web service,
- A User that wants to access the service,
- A Service Consumer that will access the Service Provider on behalf of the user.

In this context, the end user may have different accounts on each application.

The User may also want to grant more or less rights to the consumer. Typically, when Application A accesses services of Application B, it may be granted only read access. But when using Application C, users wants to grant Read/Write.



In this section

3-Legged OAuth

OAuth provides a solution to handle this problem.

The synopsis of the authentication goes like that:

1. User wants to access a module of Application A that needs to access a Service hosted by B (for example an OpenSocial Gadget).
2. Application A will request a "Request Token" from Application B.
3. Application B will respond with:
   - A Request Token,
   - An authentication URL.
4. User will be redirected to the authentication URL on the Application B.
5. User will then:
   a. Authenticate against Application B;
   b. Accept the access request from Application A displayed by Application B (and maybe specify some associated security).
6. User will then be redirected to Application A (on the callback URL, with the token and a random verifier).
7. Application A will receive the request and call Application B to exchange the Request Token for an Access Token
8. Application B will only allow the exchange if:
   - The verifier is OK,
   - The Request Token has been granted by user.
10. Application A will then:
    - Store the Access Token,
    - Use it to access Service in B on behalf of User.

The interesting part is that:

- User does not have to give Application A his login/password on Application B,
- User can choose how Application A can access his data on Application B:
  - Define security if needed,
  - He should also be able to revoke the grant at any time;
- Application A can store the token and reuse it as long as it is valid (no more authentication needed),
- Application A and B do not need to share the same user DB.

2-Legged OAuth (Signed Fetch)

Of course 3-Legged OAuth is not always the best option. In some situations, you only want to do a server-to-server authentication.

To answer this use case, OAuth supports a simple mode that will only use the OAuth signature system (see below) to have a trusted communication between two servers.

OAuth Signature

An important part of OAuth is that the HTTP requests have to be signed:

- To be able to verify the identity of the caller,
- To be sure the request was not modified during the transfer.

For that OAuth can use two different signing methods:

- HMAC symmetric signature (Shared Secret),
- RSA1 asymmetric signature (Public /Private keys).

All requests using OAuth (2-Legged or 3-Legged) are signed using one of these two signature method. The signature method is part of the request and may be chosen by the client, but the server may also enforce one of them.

In terms of security both methods have pros and cons, but RSA1 is generally more expensive.

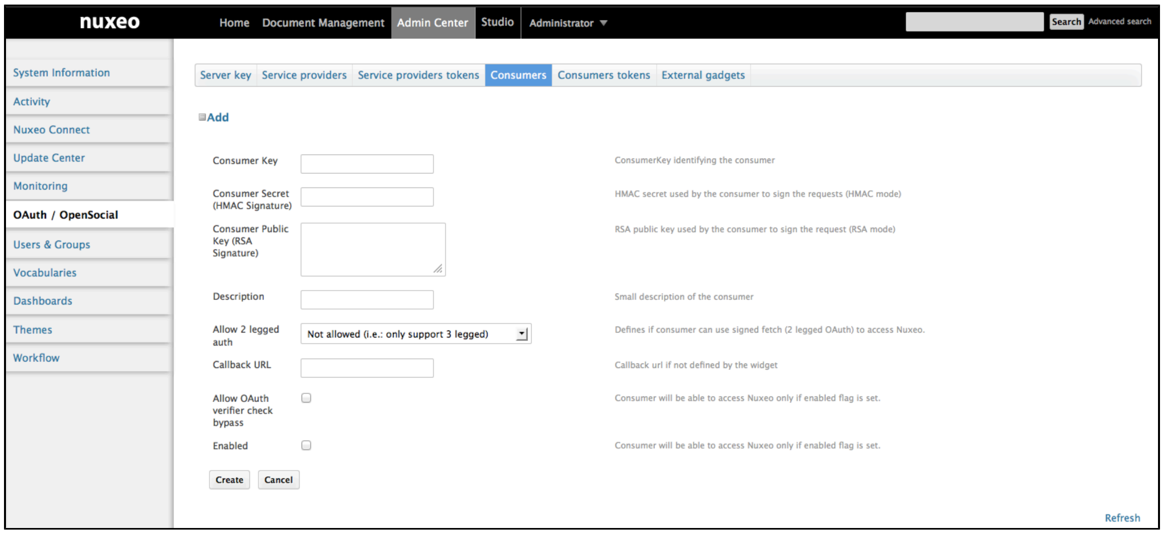
OAuth in Nuxeo Platform

Managing Service Consumers in Nuxeo

Generic Consumers

Consumers are declared in Nuxeo OAuthConsumerRegistry.

You can access this service via the **Admin Center** > **OAuth / OpenSocial** > **Consumers**.



Consumer Key		ConsumerKey identifying the consumer
Consumer Secret (HMAC Signature)		HMAC secret used by the consumer to sign the requests (HMAC mode)
Consumer Public Key (RSA Signature)		RSA public key used by the consumer to sign the request (RSA mode)
Description		Small description of the consumer
Allow 2 legged auth	Not allowed (i.e.: only support 3 legged)	Defines if consumer can use signed fetch (2 legged OAuth) to access Nuxeo.
Callback URL		Callback url if not defined by the widget
Allow OAuth verifier check bypass	☐	Consumer will be able to access Nuxeo only if enabled flag is set.
Enabled	☐	Consumer will be able to access Nuxeo only if enabled flag is set.

To define a OAuth Consumer you will need to enter:

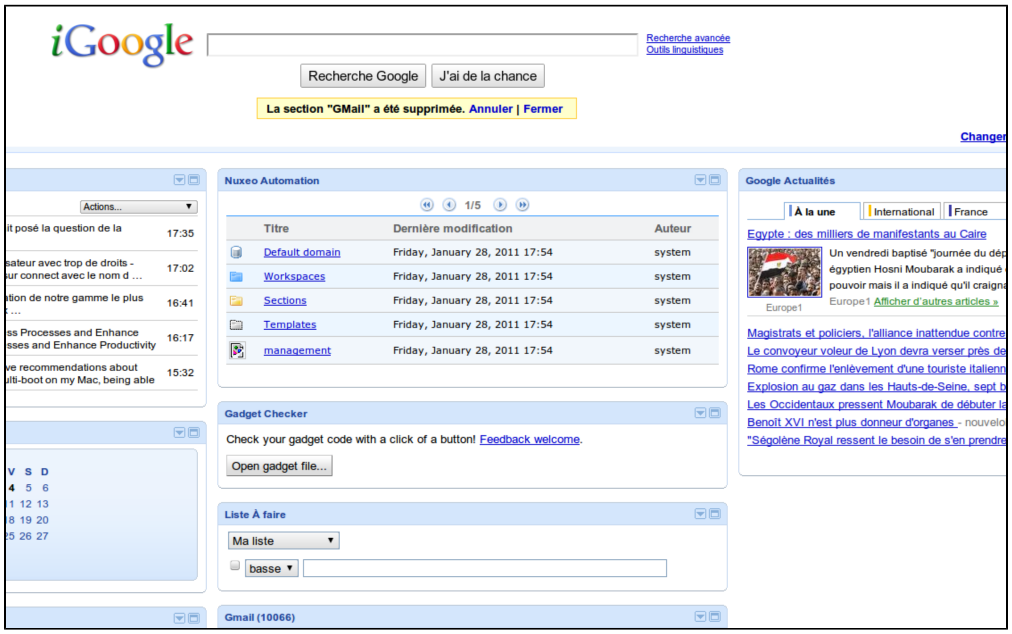
- A Consumer Key: identifier of the application,
- A Consumer Secret (HMAC key) and/or a RSA1 public key,
- A callback URL:
  - Only needed for 3-Legged OAuth,
  - Most consumers will automatically provide this callback URL as part of the OAuth dialog.
- A security mode:
  - You can restrain consumer usage to 3-Legged OAuth;
  - You can accept 2-Legged OAuth (i.e. only server-to-server authentication). In this case you will need to choose how the Nuxeo user will be chosen:
    - OpenSocial viewer,
    - OpenSocial owner,
    - Dedicated login.

Depending on the consumer, it may not allow all possibilities:

- Maybe only HMAC signature will be supported;
- Maybe only 3-Legged OAuth will be supported;
- ...

For example, you may want to use iGoogle as a consumer of some of your Nuxeo Gadgets and services. In this case, you will need to:

- Use a consumer with www.google.com as consumer key;
- Use RSA1 key as documented here;
- Provide in Nuxeo the callback URL http://oauth.gmodules.com/gadgets/oauthcallback.



La section "GMail" a été supprimée. Annuler | Fermer

	Titre	Dernière modification	Auteur
	Default domain	Friday, January 28, 2011 17:54	system
	Workspaces	Friday, January 28, 2011 17:54	system
	Sections	Friday, January 28, 2011 17:54	system
	Templates	Friday, January 28, 2011 17:54	system
	management	Friday, January 28, 2011 17:54	system

it posé la question de la	17:35
sateur avec trop de droits - ur connect avec le nom d ...	17:02
tion de notre gamme le plus ...	16:41
ss Processes and Enhance sses and Enhance Productivity	16:17
ve recommendations about lti-boot on my Mac, being able	15:32

Check your gadget code with a click of a button! Feedback welcome.

Egypte : des milliers de manifestants au Caire
Un vendredi baptisé "journée du dép
égyptien Hosni Moubarak a indiqué
pouvoir mais il a indiqué qu'il craign
Europe1 Afficher d'autres articles »
Magistrats et policiers, l'alliance inattendue contre
Le convoyeur voleur de Lyon devra verser près de
Rome confirme l'enlèvement d'une touriste italienn
Explosion au gaz dans les Hauts-de-Seine, sept b
Les Occidentaux pressent Moubarak de débuter la
Benoît XVI n'est plus donneur d'organes - nouvelo
"Ségolène Royal ressent le besoin de s'en prendre

V S D
4 5 6
1 12 13
8 19 20
5 26 27

The Special Case of Nuxeo/Shindig

When Nuxeo is both the provider and the consumer, 2-Legged OAuth (signed Fetch) is used with a HMAC Signature. The HMAC secret is dynamically generated at runtime and shared between Shindig and Nuxeo in memory.

Nuxeo OAuth URLs

Access URL: `/nuxeo/oauth/access-token` (URL to request an Access Token).

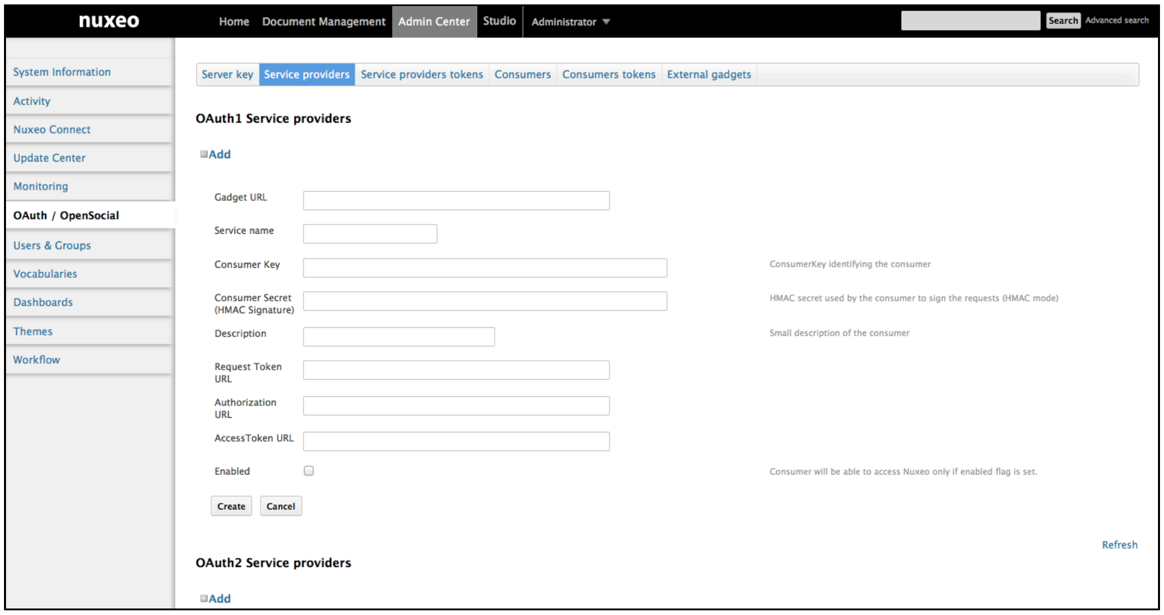
Request URL: `/nuxeo/oauth/request-token` (URL to request a Request Token).

Authorization URL: `/nuxeo/oauth/authorize` (URL used by a User to grant access).

Managing Service Providers in Nuxeo

Service providers are declared in OAuthServiceProviderRegistry.

You can access it via **Admin Center** > **OAuth / OpenSocial** > **Service providers**.



OAuth1 Service providers

Add

Gadget URL		
Service name		
Consumer Key		ConsumerKey identifying the consumer
Consumer Secret (HMAC Signature)		HMAC secret used by the consumer to sign the requests (HMAC mode)
Description		Small description of the consumer
Request Token URL		
Authorization URL		
AccessToken URL		
Enabled	☐	Consumer will be able to access Nuxeo only if enabled flag is set.

OAuth2 Service providers

Add

You will have to enter the following information:

- A gadget URL (AppId): the identifier of the gadget inside Nuxeo that will use the service,
- A service Name: the name of the service as used in the makeRequest call,
- A consumer Key: the consumer key (login) that must be used to call the service,
- A consumer Secret (HMAC key): a shared secret used to sign OAuth requests,
- OAuth callback URLs.

OAuth callback URLs will only be used if the gadget does not provide the needed information as part of the Gadget Spec (it should). If the consumer secret is empty, Nuxeo with use the server global RSA1 key.

When looking up the right service provider, Nuxeo will do the search based on the service Name and the AppId and fallback to the closest match.

Managing Tokens

OAuth requires Nuxeo to manage Tokens:

- Request Tokens are managed in memory (transient),
- Access Token are managed in the SQL DB.

You can use the Admin Center screens to:

- Add the Tokens that Nuxeo granted or was granted,
- Revoke (delete) the Tokens.

Other OAuth Documentation

Using OAuth (Nuxeo Installation and Administration - 5.8)

OpenSocial, OAuth and the Nuxeo Platform (Nuxeo Platform Developer Documentation - 5.8)

Using Shibboleth

Shibboleth has two major halves: an identity provider (IdP), which authenticates users and releases selected information about them, and a service provider (SP) that accepts and processes the user data before making access control decisions or passing the information to protected applications. These entities trust each other to properly safeguard user data and sensitive resources.

Here is the process of authentication:

1. The user accesses a protected resource.
2. The SP determines an IdP and issues the authentication request.
3. The user authenticates to the IdP.
4. The IdP issues a response to the SP.
5. The SP decodes the message and extracts the attributes.
6. The browser is redirected to the protected resource.

For details, see https://spaces.internet2.edu/display/SHIB2/FlowsAndConfig.

Authentication Plugin

The SHIB_AUTH plugin is implemented by the class `org.nuxeo.ecm.platform.shibboleth.auth.ShibbolethAuthenticationPlugin`. It authenticates the user based on HTTP headers received from the SP. It also creates (or updates) an entry in the userDirectory for this user.

In this section

As the Shibboleth attributes values are passed by HTTP headers, the service `org.nuxeo.ecm.platform.shibboleth.service.ShibbolethAuthenticationService` has been added to configure the mapping between the user metadata and the headers names.

Installation

Module Installation

The Easy Way

The Shibboleth authentication module could be installed through the Marketplace with the Shibboleth Authentication package.
See the Installing a New Package on Your Instance page.

The Manual Way

You can also download the built nuxeo-platform-login-shibboleth.jar, and deploy it into your Tomcat or JBoss instance, in the `bundles` directory.

If you did the installation by just deploying the `nuxeo-platform-login-shibboleth.jar` into your Nuxeo instance, you need to add a new configuration file to define the login and logout URLs, the mapping between the user metadata and the headers names.

Add a new file named `shibboleth-config.xml` in the `config/` directory of your server.

- `$NUXEO/nxserver/config` for a Tomcat distribution
- `$NUXEO/server/default/deploy/nuxeo.ear/config` for a JBoss distribution

shibboleth-config.xml

```
<?xml version="1.0"?>
<component name="org.nuxeo.ecm.platform.login.shibboleth.config">
<extension
    target="org.nuxeo.ecm.platform.shibboleth.service.ShibbolethAuthenticationService"
    point="config">
    <config>
      <uidHeaders>
        <default>uid</default>
      </uidHeaders>

      <loginURL>https://host/Shibboleth.sso/WAYF</loginURL>
      <logoutURL>https://host/Shibboleth.sso/Logout</logoutURL>

      <fieldMapping header="uid">username</fieldMapping>
      <fieldMapping header="mail">email</fieldMapping>
    </config>
  </extension>
</component>
```

Overriding the Default Authentication Chain

To enable the Shibboleth authentication, you need to add the Shibboleth plugin to the authentication chain.

To override the default authentication chain in Nuxeo DM, add a new file named `authentication-chain-config.xml` in the `config/` directory of your server.

authentication-chain-config.xml

```
<?xml version="1.0"?>
<component name="org.nuxeo.ecm.platform.your.authentication.chain.config">
  <require>org.nuxeo.ecm.platform.ui.web.auth.WebEngineConfig</require>
  <extension
target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
    point="chain">
    <authenticationChain>
      <plugins>
        <plugin>BASIC_AUTH</plugin>
        <plugin>SHIB_AUTH</plugin>
        <plugin>FORM_AUTH</plugin>
        <plugin>WEBENGINE_FORM_AUTH</plugin>
        <plugin>ANONYMOUS_AUTH</plugin>
        <plugin>WEBSERVICES_AUTH</plugin>
      </plugins>
    </authenticationChain>
  </extension>
</component>
```

If you already defined your own authentication chain in any of your config or contrib files, you just need to add the `SHIB_AUTH` plugin into your own chain.

Anonymous Authentication Compatibility

As it is not possible to write access rules based on resource URLs with Nuxeo, Shibboleth LazySession has to be enabled. By adding the ANONYMOUS_AUTH in the authentication chain, the Shibboleth login will only be asked when accessing a restricted resource.

For that, the `org.nuxeo.ecm.platform.shibboleth.auth.exceptionhandling.ShibbolethSecurityExceptionHandler` will redirect to the login URL when a NuxeoSecurityException is thrown while the current user is anonymous.

To activate it, add a new file named `login-anonymous-config.xml` in the `config/` directory of your server.

login-anonymous-config.xml

```
<?xml version="1.0"?>
<component name="org.nuxeo.ecm.platform.your.anonymous.user.config">
  <extension target="org.nuxeo.ecm.platform.usermanager.UserService"
    point="userManager">
    <userManager>
      <users>
        <anonymousUser id="Guest">
          <property name="firstName">Guest</property>
          <property name="lastName">User</property>
        </anonymousUser>
      </users>
    </userManager>
  </extension>
</component>
```

Full Sample Configuration File

Here is a sample configuration file containing everything you need to set up the Shibboleth authentication module:

```
<component name="sample.shibboleth.config">

  <require>org.nuxeo.ecm.platform.ui.web.auth.WebEngineConfig</require>
  <require>org.nuxeo.ecm.platform.usermanager.UserManagerImpl</require>

  <extension

target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
      point="chain">
    <authenticationChain>
      <plugins>
        <plugin>BASIC_AUTH</plugin>
        <plugin>SHIB_AUTH</plugin>
        <plugin>ANONYMOUS_AUTH</plugin>
      </plugins>
    </authenticationChain>
  </extension>

  <extension
    target="org.nuxeo.ecm.platform.shibboleth.service.ShibbolethAuthenticationService"
    point="config">
    <config>
      <uidHeaders>
        <default>uid</default>
      </uidHeaders>

      <loginURL>https://host/Shibboleth.sso/WAYF</loginURL>
      <logoutURL>https://host/Shibboleth.sso/Logout</logoutURL>

      <fieldMapping header="uid">username</fieldMapping>
      <fieldMapping header="mail">email</fieldMapping>
    </config>
  </extension>

  <!-- Add an Anonymous user -->
  <extension target="org.nuxeo.ecm.platform.usermanager.UserService"
    point="userManager">
    <userManager>
      <users>
        <anonymousUser id="Guest">
          <property name="firstName">Guest</property>
          <property name="lastName">User</property>
        </anonymousUser>
      </users>
    </userManager>
  </extension>

</component>
```

Source Code

The source code of the Shibboleth authentication module can be found as part of the `nuxeo-platform-login` addon on GitHub

ShibbGroups Addon

ShibbGroups are virtual groups based on an EL expression with Shibboleth attributes. A new user management tab is added to create and edit them. The definitions are stored in the `shibbGroup` directory.

The class `org.nuxeo.ecm.platform.shibboleth.computedgroups.ShibbolethGroupComputer` computes at login time the ShibbGroups that the current user is member of.

The source code of this addon can be found on GitHub.

Using Kerberos

Here's an How to to help you configure the SPNEGO/Kerberos authentication for Nuxeo. Note that this it starts with OS relative guidelines.

Configuring Kerberos on Linux

1. Configure Kerberos for your server and client. For example in a Debian-based Linux server install `krb5-kdc` and `krb5-admin-server`, and setup a realm (with krb5_newrealm).
2. Create a service principal and set its service principal name to `HTTP/@REALM`.
3. Export the service principal keytab In MIT Kerberos. Using kadmin, type the following commands:

```
add_principal HTTP/servername (type in a
password)
```

Principal format
The service principal MUST be formed as followed: uppercase HTTP slash the canonical (DNS-wise) name of the server. Any other names will not work (especially, aliases).

```
ktadd -k /tmp/keytab HTTP/servername
```

You may create as many principals you want and add their keys to the same keytab, e.g. `HTTP/nuxeo@COMPANY` and `HTTP/nuxeo.company.com@COMPANY`.

In this section

Configuring Kerberos on Windows

1. In Microsoft Active Directory, create a user. Option "password does not expire" must be checked and "user must change password" unchecked.
2. In a command-line window, register the service principal name(s) you want this user to respond to (generally the server name in its short and fully qualified forms).

Service principal name format
The service principal *has* to correspond to the server's canonical name in DNS.

```
setspn -a HTTP/servername@REALM setspn -a
HTTP/fully.qualified.servername@REALM
```

3. Check the SPNs with the command:

```
setspn -l
```

4. Export the keytab.

```
ktpass /out C:\temp\http.keytab /princ HTTP/servername@REALM /mapuser
domain\username /pass password /crypto RC4-HMAC-NT /ptype
KRB5_NT_PRINCIPAL /kvno 0
```

`/mapuser` and `/pass` should not be strictly necessary after `setspn`, but better be safe than sorry.

Generic

1. Copy the `keytab` on your Nuxeo server.
2. Configure krb5.conf (`/etc/krb5.conf` or `C:\Windows\krb5.ini`, depending on the OS).

   Note that on Linux servers, while it is not strictly necessary, you should really install the MIT Kerberos user tools (krb5-user in Debian-like). This will setup a basic `krb5.conf` and give you debugging tools.

   The `krb5.conf` should minimally contain:

```
[libdefaults]
        default_realm    REALM
     [realms]
          REALM = {
              kdc = <kdc>
              admin_server = <admin_server>
          }
     [domain_realm]
          domain = REALM
          .domain = REALM
```

   - `kdc` and `admin_server` are the names of your kdc and admin servers (duh). On Windows, both are your AD server. On Linux, `kdc` is where you've installed `krb5-kdc` and `admin_server` in where you've installed `krb5-admin-server`. Usually it's the same machine.
   - `domain` is the DNS domain you want to map to a realm. In Linux clients, specifying the realm is necessary. It's not on Windows, but again, better be safe than sorry.
3. Test the Kerberos installation. On Linux servers, you can test it with the command:

```
kinit -k -t /path/to/keytab HTTP/servername@REALM
```

   There should be no errors and klist should list a krbtgt for your service. On my machine that looks like this:

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: HTTP/loremipsum@LOREMIPSUM
Valid starting     Expires            Service principal
15/12/12 11:35:36  15/12/12 21:35:36  krbtgt/LOREMIPSUM@LOREMIPSUM
  renew until 16/12/12 11:35:36
```

4. If you use your server as a Kerberos client too (e.g. it's a development machine!), delete the tgt with the command `kdestroy`.

Configuring Java

To enable Kerberos, you need to use a login configuration implementation. You have two ways of doing this. Either change the default Java configuration or use JAVA_OPTS. Take a look at JAAS documentation for more details. If you have installed the Marketplace package, the JAVA_OPTS is automatically added.

Changing the default JRE configuration

1. In `$JAVA_HOME/jre/lib/security/java.security`, find the login config url (it's commented out by default):

   ```
   #login.config.url.1=file:${user.home}/.java.login.config
   ```

2. Set this to a regular file, e.g. `/opt/nuxeo/java.login.config`.

Giving a Custom Login File as Java Argument

In `nuxeo.conf`, add the following line:

```
JAVA_OPTS=$JAVA_OPTS -Djava.security.auth.login.config=./java.login.config
```

Note that using one equal sign appends or overrides parts of the default `java.security` file, whereas using two equal signs completely overrides the default `java.security` file content.

```
JAVA_OPTS=$JAVA_OPTS -Djava.security.auth.login.config==./java.login.config
```

Configuring JAAS

If you have installed the Marketplace package, this file is already available at `$NUXEO_HOME/java.login.config`.

Open the `java.login.config` file you've setup and add the following configuration:

```
Nuxeo {
  com.sun.security.auth.module.Krb5LoginModule required
  debug=true
  storeKey=true
  useKeyTab=true
  keyTab="/complete/path/to/keytab"
  principal="HTTP/servername@REALM";
 };
```

Login configuration name
The login configuration MUST be called Nuxeo with an uppercase N.

Configuring Nuxeo (We're Almost There!)

1. Deploy the bundle in `$NUXEO_HOME/nxserver/bundles`.
2. Create a `$NUXEO_HOME/nxserver/config/kerberos-config.xml` with the following content:

```
<?xml version="1.0"?>
 <component name="Kerberos-config">
  <require>org.nuxeo.ecm.platform.ui.web.auth.WebEngineConfig</require>
  <require>org.nuxeo.ecm.platform.ui.web.auth.defaultConfig</require>
  <require>org.nuxeo.ecm.platform.login.Kerberos</require>
  <documentation> This Authentication Plugin uses Kerberos to assert user
identity. </documentation>
  <extension
target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService
" point="authenticators">
   <authenticationPlugin name="KRB5_AUTH" enabled="true"
class="org.nuxeo.ecm.platform.ui.web.auth.krb5.Krb5Authenticator">
    <loginModulePlugin>Trusting_LM</loginModulePlugin>
   </authenticationPlugin>
  </extension>
  <extension
target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService
" point="chain">
   <authenticationChain>
    <plugins>
     <plugin>BASIC_AUTH</plugin>
     <plugin>KRB5_AUTH</plugin>
     <plugin>FORM_AUTH</plugin>
    </plugins>
   </authenticationChain>
  </extension>
  </component>
```

For now we assume all configuration - realm, kdc, server principal, etc. lives in the server's standard configuration, i.e. either `java.login.config` or `krb5.conf`. An interesting update would be to make these configurable in Nuxeo.

3. Start Nuxeo.

Configuring Client

On Windows:

- If the client is IE or Chrome, no further configuration should be necessary. Jjust ensure your Nuxeo server is in the Local intranet or Trusted sites security zone.
- If the client is Firefox, go to about:config, search for `network.negotiate-auth.trusted-uris` and set it to your full server URL.

On Linux, if the client is Firefox:

1. Go to about:config.
2. Search for `network.negotiate-auth.trusted-uris` and set it to your full server URL.
3. Configure `krb5.conf` as above.
4. Get a Kerberos ticket with `kinit`.

In the browser, type http://nuxeo_server:8080/nuxeo and… you should be authenticated!

⚠ **URL**
Do not use localhost in the URL but the server's canonical name, that you mapped to the SPN.

Note

The authenticator supports a magic request header to disable it. Simply set the `X-Skip-Kerberos` request header and Nuxeo will move on to the next filter on the list. This is useful if you want integrated authentication from within a corporate network but not from outside: simply setup two Apache virtual hosts with an "internal" URL and an "external" URL. In the external virtual host, add the following directive:

```
RequestHeader set X-Skip-Kerberos true
```

and this will move on to form authentication.

HTTP and HTTPS Reverse-Proxy Configuration

The Nuxeo webapp can be virtual hosted behind a HTTP/HTTPS reverse proxy, like Apache, NGINX, IIS, etc.

Motivations for Virtual Hosting

Virtual hosting provides several advantages:

- Support for HTTPS
  HTTPS support in Apache is easy and flexible to setup. Apache can also be used to handle certificate authentication.
- URL filtering
  You can use Apache filtering tools to limit the URLs that can be accessed via the reverse proxy.
- Handle HTTP cache for static resources
  The Nuxeo Platform generates standard HTTP cache headers for all static resources (images, JavaScript...). These resources are by default cached on the client side (in the browser cache). For performance reasons, it can be useful to host these resources in the reverse proxy cache.

In this documentation page, focus is on the Apache Configuration. Other configurations are available as children of this page:

- Configuring a Reverse Proxy to Work with Live Edit and Client Certificate Authentication
- Internet Information Services (IIS) Configuration

In this section

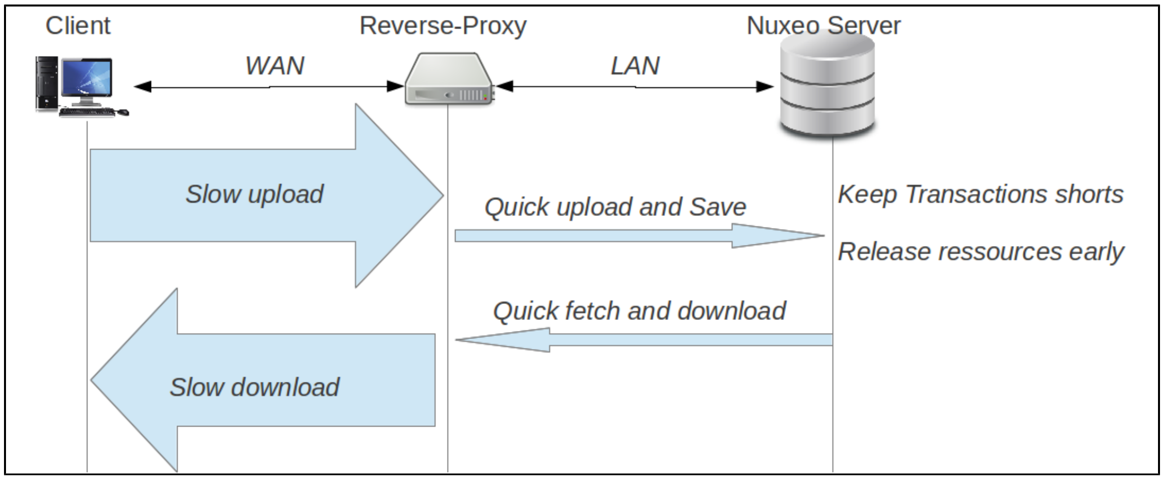
Bandwidth and Transactions Optimizations

Using a reverse proxy can also be a wait to optimize uploads and downloads.

When clients access the Nuxeo server via a WAN with low bandwidth, upload and download can take time: this is bad for the server since it consumes resources:

- HTTP socket and HTTP thread,
- memory,
- transaction and associated JDBC resources.

A reverse proxy can be used to buffer uploads and downloads so that uploads and downloads from the Nuxeo server's point of view are always done at a high speed.



Virtual Hosting Configuration for Apache 2.x

Reverse Proxy With mod_proxy

For this configuration, you will need to load and enable the `mod_proxy` and `mod_proxy_http` modules.

```
ProxyPass /nuxeo/ http://Nuxeo5ServerInternalIp:8080/nuxeo/
ProxyPassReverse /nuxeo/ http://Nuxeo5ServerInternalIp:8080/nuxeo
ProxyPreserveHost On
```

You can also use rewrite rules to achieve the same result:

```
ProxyVia On
ProxyRequests Off
RewriteEngine On
RewriteRule /nuxeo(.*) http://Nuxeo5ServerInternalIp:8080/nuxeo$1 [P,L]
```

This configuration will allow you to access the Nuxeo Platform webapp via http://ApacheServer/nuxeo/.

The Nuxeo webapp will generate the URLs after reading the http header x-forwarded-host.

Unfortunately, this header does not specify the protocol used. So if your Apache is responding to HTTPS, you will need to send the Nuxeo Platform a specific header to indicate the base URL that the webapp must use when generating links.

```
RequestHeader append nuxeo-virtual-host "https://myDomainName/"
```

This will require you to load and activate the `mod_headers` module.

And if you have a "client denied by server configuration" error, you must check the access control directives of `mod_proxy`:

```
<Proxy *>
  Order Deny,Allow
  Deny from all
  Allow from 192.168
</Proxy>
```

Reverse Proxy a WebEngine Site to a myexample.com/mysite URL

You need the same configuration from the first section. It is advised to first get it to work before proxying exclusively a Webengine site.

A site request queries both from its own URL (/nuxeo/site/mysite) but also gets static resources from the root (/nuxeo/nxthemes ...).
A rewrite configuration for mysite would look like:

```
RewriteRule ^/nuxeo$    /nuxeo/   [P,L]
RewriteRule ^/mysite$    /mysite/   [P,L]

RewriteCond %{REQUEST_URI}  ^/mysite/skin/mysite/.*
RewriteRule ^/mysite/skin/mysite/(.*)
http://127.0.0.1:8080/nuxeo/site/skin/mysite/$1 [P,L]

RewriteCond %{REQUEST_URI}  ^/mysite/skin/.*
RewriteRule ^/mysite/skin/(.*) http://127.0.0.1:8080/nuxeo/site/skin/mysite/$1 [P,L]

RewriteCond %{REQUEST_URI}  ^/mysite/nxthemes-(lib|css)/.*
RewriteRule ^/mysite/(.*) http://127.0.0.1:8080/nuxeo/$1 [P,L]

RewriteCond %{REQUEST_URI}  ^/mysite/nxthemes-webwidgets/.*
RewriteRule ^/mysite/(.*) http://127.0.0.1:8080/nuxeo/site/$1 [P,L]

RewriteRule ^/mysite/(.*) http://127.0.0.1:8080/nuxeo/site/mysite/$1 [P,L]
```

Webengine also needs to know the base of the site, in this case, an empty string instead of /nuxeo/site. This information is passed using the `mod_headers`:

```
RequestHeader append nuxeo-webengine-base-path ""
```

You can also fetch the static resources from a different path. To do so add a properties to the `nuxeo.properties` file:

```
org.nuxeo.ecm.webengine.skinPathPrefix=/skin/
```

Reverse Proxy With mod_jk

The `AJP` connector may lock threads if you're not using the `APR` implementation. Please read the native tomcat documentation for activating the `APR` implementation on your system. On Linux you just have to install the package `libtcnative-1`.

`mod_jk` allows you to communicate between Apache and Tomcat via the ajp1.3 protocol.

```
JkWorkersFile /etc/apache2/jk/workers.properties
JkLogFile     /var/log/mod_jk.log
JkLogLevel    info
JkMount /nuxeo ajp13
JkMount /nuxeo/* ajp13
```

The `workers.properties` file will contain the list of Nuxeo EP Tomcat servers. The AJP13 Tomcat listener should be enabled by default on port 8009.

```
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=Nuxeo5ServerInternalIp
worker.ajp13.type=ajp13
worker.ajp13.socket_keepalive=1
worker.ajp13.connection_pool_size=50
```

Once again, if you use HTTPS, you will need to set the Nuxeo-specific header to tell the webapp how to generate URLs:

```
RequestHeader append nuxeo-virtual-host "https://myDomainName/"
```

This will require you to load and activate the mod_header module.

Configuring HTTP Cache

The Simple cache filter is deprecated, we recommend using the `filterConfig` extension point of `RequestControllerService` .

RequestControllerService's Filter Extension Point

This XP lets you contribute customized filter for a given pattern URL.

Example of a filterConfig Registration

```
<extension
target="org.nuxeo.ecm.platform.web.common.requestcontroller.service.RequestControlle
rService"
  point="filterConfig">

   <filterConfig name="filterName" transactional="true" synchonize="true"
     cached="true" private="true" cachetime="3600">
   <pattern>/nuxeo/urlPattern/.*</pattern>
   </filterConfig>
 </extension>
```

This contribution will ensure that every pattern matching URL will go through NuxeoRequestControllerFilter. The header of the corresponding request will be modified according to the XP configuration. Here is a list of the possible options:

- `filterConfig`
  - name: name of the Filter.
  - transactional: use transaction.
  - synchonize: is synchronized.
  - cached: if true, adds cache-control to the header.
  - cacheTime: cache duration.
  - private: if true, cache is private, public if false.
- `pattern`: URL pattern to match

Using Simple Cache Filter

The Nuxeo webapp includes a servlet filter that will automatically add header cache to some resources returned by the server.

By using the `deployment-fragement.xml` you can also put some specific resources behind this filter:

```
<extension target="web#FILTER">
  <filter-mapping>
    <filter-name>simpleCacheFilter</filter-name>
    <url-pattern>/MyURLsToCache/*</url-pattern>
  </filter-mapping>
</extension>
```

When the Nuxeo Platform is virtual hosted with apache you can use `mod_cache` to use the reverse-proxy as cache server.

You can also use Squid or any other caching system that is compliant with the standard HTTP caching policy.

Nuxeo Tomcat HTTPS Configuration

Configuring Tomcat in HTTPS is not recommended. Please follow instructions above to configure Apache server.

If you need to configure your Nuxeo Tomcat in HTTPS, the platform provides an `HTTPS` configuration template for this purpose:

Add `https` to the `nuxeo.templates` property then edit the related properties.

Sample configuration of properties related to the "https" template

```
nuxeo.server.https.port=8443
nuxeo.server.https.keystoreFile=/path/to/keystore
nuxeo.server.https.keystorePass=password
```

To create your keystore using Java
keytool -genkey -alias tomcat -keyalg RSA

Related pages

- HTTP and HTTPS Reverse-Proxy Configuration (Nuxeo Installation and Administration - 5.8)
- Configuring a Reverse Proxy to Work with Live Edit and Client Certificate Authentication (Nuxeo Installation and Administration - 5.8)
- Navigation URLs (Nuxeo Platform Developer Documentation - 5.8)
- URLs for Files (Nuxeo Platform Developer Documentation - 5.8)
- Default URL Patterns (Nuxeo Platform Developer Documentation - 5.8)
- Default WebEngine Applications (Nuxeo Platform Developer Documentation - 5.8)
- Downloading Files (Nuxeo Platform Developer Documentation - 5.8)
- REST API (Nuxeo Platform Developer Documentation - 5.8)

Configuring a Reverse Proxy to Work with Live Edit and Client Certificate Authentication

The configuration below has been tested and was found to work on:

- Server: Ubuntu Server 12.04 LTS + Nuxeo Platform 5.6,
- Ubuntu Client: Ubuntu Desktop 12.04 LTS + Firefox + Nuxeo LiveEdit Protocol Handler 0.5.2,
- Windows Client: both Windows 7 and Windows XP + Nuxeo LiveEdit plugin for IE.

To configure a reverse proxy to work with Live Edit:

1. After installing Apache server, enable site SSL and necessary modules using the commands below:

```
a2ensite default-ssl

a2enmod ssl proxy proxy_http headers rewrite

service apache2 restart
```

2. Create a directory called `access_control` in `/etc/apache2/` and put control directives into any file in that directory, for example:

```
SSLRequire %{SSL_CLIENT_S_DN_Email} in
{"email_address_of_allowed_user@allowed.com"}
SSLRequire  %{SSL_CLIENT_S_DN_O} in {"Allowed Organization"}
```

   This directory will be used in the configuration file.
3. Create a directory called `certs` in `/etc/ssl/` and put the CA certificates into that directory.
4. In a terminal, go to the directory `/etc/ssl/certs` and execute "`c_rehash .`" to create the required symbolic links.
5. Edit the site configuration file (`/etc/apache2/sites-enabled/default-ssl`) with the content below.
   This configuration enables reverse proxy through HTTPS, and also enables authentication by client certificate.

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerName nutest.test.com
        ServerAlias nutest.test.com
        ProxyPass / http://127.0.0.1:8080/
        ProxyPassReverse / http://127.0.0.1:8080/
        RequestHeader append nuxeo-virtual-host "https://nutest.test.com/"
        ServerAdmin webmaster@localhost
        <ProxyMatch
^http\://127\.0\.0\.1\:8080(?!((/nuxeo/restAPI/)|(/nuxeo/nxliveedit.face)))>
                SSLRequireSSL
                Include /etc/apache2/access_control
                SSLCACertificatePath /etc/ssl/test_certs/
                SSLVerifyClient optional
                SSLVerifyDepth  3
                RewriteEngine       on
                RewriteCond     %{SSL:SSL_CLIENT_VERIFY} !=SUCCESS
                RewriteRule     .? - [F]
                ErrorDocument 403 "ACCESS DENIED: You need a client side
certificate issued by EAST IP to access this site"
        </ProxyMatch>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
        SSLEngine on
        SSLCertificateFile    /etc/ssl/certs/nutest.pem
        SSLCertificateKeyFile /etc/ssl/private/nutest.key
        BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shutdown \
                downgrade-1.0 force-response-1.0
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
</VirtualHost>
</IfModule>
```

`ServerName` and `ServerAlias` must be set for LiveEdit to work on certain Java versions. See http://answers.nuxeo.com/questions/460

9/nuxeo-live-edit-throws-a-java-npe.

`ProxyPass`, `ProxyPassReverse` and `RequestHeader` directives are standard setup to enable reverse proxy. See the Reverse proxy with `mod_proxy` section.

`SSLCertificateFile` and `SSLCertificateKeyFile` provide the server certificate and private key.

About the `<ProxyMatch...>` block:

- It enables client certificate authentication. The regular expression in this directive matches PROXYED resource, which includes all but two resources: `/nuxeo/restAPI/` and `/nuxeo/nxliveedit.face`. The reason for exclusion of the two resources is that the LiveEdit plugin will not provide client certificate for server to verify. Therefore, to walk around this problem, server do not request client certificate for resources accessed by the LiveEdit plugin.
- `SSLRequireSSL` requires SSL connection.
- `Include /etc/apache2/access_control` includes files in the configured directory into this configuration. Included files check up the client certificates.
- `SSLCACertificatePath` specifies the directory where the trusted CA certificates are placed. Only client certificates issued by the trusted CAs can be accepted. Alternatively, if you use just a few certificates, you can use `SSLCACertificateFile` directiv e instead of `SSLCACertificatePath`.
- `SSLVerifyDepth` specifies the depth of trust link. Increase the number if the trust link is long.
- `SSLVerifyClient` is set to "optional" to allow the `RewriteEngine` to work, and, with the rest of directives, provide a better error message when client certificate is rejected.

The rest of the configuration is from the standard configuration template.

6. On the client side, import the client certificate into your web browser and try to access and log in to Nuxeo. If it does not work, check the reverse proxy and client certificate authentication settings, as well as the log files on server (`/var/log/apache2/ssl_access.log`).
7. If the client uses LiveEdit, and the issuer of the client certificate is not trusted by the Java Runtime Environment (JRE) on client end in which the LiveEdit plugin runs, import the issuer's certificate into the JRE's cacerts keyring with the JRE's keytool:

```
keytool -importcert -trustcacerts -alias alias_for_your_ca -file your_ca_cert.pem
-keystore /opt/jdk1.7.0_09/jre/lib/security/cacerts
```

If in any case LiveEdit does not work and throw a Java exception, do the follows to diagnose the problem:

1. Find the log file for the plugin.
   For Firefox, look at "`Tools` > `Add-ons` > `Nuxeo LiveEdit Protocol Handler` > `Preferences`" and find the working directory.
   For Windows, search for the log file under the user directory. The log file may exist only AFTER the Java exception is thrown and named "`nuxeo-liveedit-openoffice-extension.log`".
2. Inspect the log file and search for solution.
   The log file may contain the complete stack dump and other information to help to diagnose the problem.

Related topics

- Configuring a Reverse Proxy to Work with Live Edit and Client Certificate Authentication (Nuxeo Installation and Administration - 5.8)
- LiveEdit icons are still available in Nuxeo after LiveEdit has been uninstalled (Nuxeo Technical Knowledge Base (FAQ))
- LiveEdit makes MS Office slow to start (Nuxeo Technical Knowledge Base (FAQ))
- Setup Firefox protocol handler with LiveEdit 2 for MS Office and OpenOffice.org (Nuxeo Technical Knowledge Base (FAQ))
- Installing Live Edit Silently (Nuxeo Installation and Administration - 5.8)
- I can't view my websites and blogs (displays a message "The HTTP header field "Accept" with value...") (Nuxeo Technical Knowledge Base (FAQ))
- Managing Your Own File with LiveEdit (Nuxeo Platform User Documentation - 5.8)
- Installing Live Edit (Nuxeo Platform User Documentation - 5.8)
- Live Edit Compatibility Table (Nuxeo Platform User Documentation - 5.8)
- HTTP and HTTPS Reverse-Proxy Configuration (Nuxeo Installation and Administration - 5.8)
- Configuration Examples (Nuxeo Installation and Administration - 5.8)
- Working with Live Edit (Nuxeo Platform User Documentation - 5.8)

Internet Information Services (IIS) Configuration

This documentation is a draft version as it was never tested in production. Please give us feedback, we will take care of your questions about this configuration. You can also share configuration improvements with the community on answers.nuxeo.com.

This documentation gives you the guidelines to install a Nuxeo instance on a Windows Server and use IIS as a frontal web server. This documentation is more focused on the IIS configuration. For more details about the installation of Nuxeo or IIS, please refer to the relevant documentation.

Requirements:

- Windows 2008 or greater,
- Java 6 or above (for Nuxeo 5.6) and Java 7 or above (for Nuxeo 5.7.1 and greater),
- IIS 7 or above with ASP.NET role service enabled,
- Nuxeo 5.6 or greater installed.

Configuration

Nuxeo Installation

1. Check Java is correctly installed.
2. Download the Nuxeo Windows distribution (.exe).
3. Install the Nuxeo Platform.
4. Start the Nuxeo instance.
5. Open the Browser from the Windows Server (firewalls must be enabled) at the address localhost:8080/nuxeo.
6. Configure the server, but in General Settings, replace IP Address 0.0.0.0 by 127.0.0.1 to limit Tomcat to local answers.

In this section

Enabling Web Server (IIS)

1. Connect as Administrator on the Windows Server.
2. In the Server Manager, click on **Add Roles and Features**.
3. Select **Web Server (IIS)**.
4. Leave the default configuration and validate.

Activating the Application Request Routing Add-On

1. Download the "Application Request Routing" addon.
2. Run the command:

   ```
   net stop was /y
   ```

   It might not be started, so ignore errors here.
3. Run the command:

   ```
   net stop wmsvc /y
   ```

   It might not be started, so ignore errors here.
4. Execute the downloaded .exe file and accept the default configuration.

If you have errors during the "Application Request Routing" installation

- Check that Windows Service Pack is up-to-date (**Start** > **Settings** > **Control Panel** > **Automatic Updates**).

- Also check the requirements (Windows version, etc.).

Configuring IIS

Enabling the Rewrite Feature

1. Go into the **Administration Tools** panel.
2. Open **Internet Information Service (IIS) Manager**.
3. Select the local server node on the left panel.
4. Click on "Application Request Routing" and open the feature.
5. Edit the server proxy settings:
    a. Check the "Enable Proxy" box.
    b. Leave the default values.
6. Save the configuration.

Adding the Rewrite Rule

1. In Internet Information Service (IIS) Manager, select a server node on the left panel.
2. Open **URL Rewrite**.
3. Click on **Add a rule(s)...**.
4. Select **Inbound rule** > **Blank rule**.
   In the following steps, if a field is not specified, leave the default value.
5. In **Match URL**:
    - Name: Nuxeo Inbound
    - Request URL: Match the pattern
    - Pattern: `(nuxeo.*)`
6. In **Server Variables**:
    - Server Variable Name: `HTTP_NUXEO_VIRTUAL_HOST`
    - Value: `{HTTP_HOST}`
    - Check the box.
7. In **Action**:
    - Action type: Rewrite
    - `http://127.0.0.1:8080/{R:1}`

If your Nuxeo server is not hosted by the IIS server, replace 127.0.0.1 by the internal IP address of the Nuxeo server. Of course, IIS server must reach this internal IP Address.

Preserving the Host Header

We've had report that the following is needed as well:

```
appcmd.exe set config -section:system.webServer/proxy -preserveHostHeader:true
```

Enhancing the URL MAX Length Parameter

Windows Web Server defines a default limit for URL paramaters. This limit is set to 256 characters. In Nuxeo, parameter length can be greater than this default value.

1. Open the Window registry: Command + R, and type `regedit`.
2. Open `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\HTTP\Parameters`.
3. Right-click on **Parameters** > **New** > **DWORD**.
    - Name: `UrlSegmentMaxLength`
    - Value: 2048
    - Base: Decimal
4. Restart the server.

Testing Your Configuration

We suggest to use a client desktop with Firefox and Firebug installed.

1. Open Firebug.
2. In Firebug, click on the **Net** section.
3. Enable the Net feedback.

4. Go to http://iis.servername/nuxeo, where `iis.servername` is the name of the server the client can reach.
5. Connect as Administrator to Nuxeo.
   Each line displayed is a request made by the browser.
6. Check that each line points to the `iis.servername` and not the 127.0.0.1 address without error (no red lines).

If you have errors, please check the next section.

Typical Errors

Error 1

> *502 - Web server received an invalid response while acting as a gateway or proxy server.*
>
> *There is a problem with the page you are looking for, and it cannot be displayed. When the web server (while acting as a gateway or proxy) contacted the upstream content server, it received an invalid response from the content server.*

In the rule you defined the target is http://internal.address.ip:8080. This error might occur because the server can't reach this address.

Error 2

> *500 - Internal server error.*
>
> *There is a problem with the resource you are looking for, and it cannot be displayed.*

This error means you made a mistake during the rule configuration and about the server variable name.

Error 3

If Nuxeo has a strange behavior, for instance:

- you can't write in a text field,
- every time you try to type something into a text field, the focus changes to the search field,
- the Home Dashboard is empty
- etc…

These problems may occur because you didn't enhance the URL MAX Length parameter.

You can look into your firebug if some requests returned a 401, with a "bad url" message.

Advanced Configuration

The pages below are about advanced configurations that are not mandatory and are used to integrate new features or change low level behaviors.

- Adding Custom Templates
- Changing Context Path
- Configuring User & Group Storage and Authentication
- Creating a Nuxeo Static WAR
- Firewall Consideration
- JDBC Datasource Configuration
- Nuxeo Clustering Configuration
- Redis Configuration
- VCS Configuration

Adding Custom Templates

The "custom" template folder allows you to add customization such as using multiple databases, configuring services, ...

1. Add your own template files in `templates/custom` directory.
   You can use either existing or new parameters in these new template files.
2. From the Admin Center or by manually editing the `nuxeo.conf` file, set your parameters' values and set `nuxeo.templates=custom`.
   You can refer to custom templates directory with a relative path or to your own custom templates directory with an absolute path.
3. Edit `custom/nuxeo.defaults` and set `nuxeo.template.includes` parameter to define the list of existing templates to include (comma separated values); your custom template will be used at last.
   `nuxeo.defaults` files from included templates are also read.

In case you need multiple customizations, create multiple directories and reference them in a dedicated `nuxeo.conf` for each server.

Other Documentation About Templates

Adding Custom Templates
Configuration Templates

Connecting Nuxeo to the Database

Configuration Parameters Index (nuxeo.conf)

Changing Context Path

Nuxeo offers the capability to change the context path, i.e. `/nuxeo` in the URL of your application.

Restriction
This documentation is about the Tomcat distribution.

This configuration is done with two steps:

1. Edit the configuration file `nuxeo.conf` to change the property `org.nuxeo.ecm.contextPath`.

```
org.nuxeo.ecm.contextPath=/myapp

# if you have configured nuxeo.url, update it as well
nuxeo.url=http://localhost:8080/myapp
```

2. Rename the file `$NUXEO_HOME/templates/common-base/conf/Catalina/localhost/nuxeo.xml.nxftl` into `$NUXEO_HOME/templates/common-base/conf/Catalina/localhost/myapp.xml.nxftl`

If you have already started Nuxeo with the old context path, you have to remove `$NUXEO_HOME/conf/Catalina/localhost/nuxeo.xml`.

Configuring User & Group Storage and Authentication

The Nuxeo Platform provides large possibilities about User and Group storage, you can:

- Bind the Nuxeo Platform users and group to ones defined into your LDAP,
- Bind the Nuxeo Platform users and group to ones defined into SQL tables (as default configuration),
- Bind the Nuxeo Platform users to ones defined into your LDAP and groups to ones locally stored into a SQL table,
- Bind the Nuxeo Platform users to ones defined into an aggregation of 2 LDAP servers and groups to ones locally stored into a SQL table,
- Bind the Nuxeo Platform users to ones defined into an aggregation of your LDAP server and a local SQL tableand groups to ones locally stored into a SQL table
- ...

Many other more complex possibilities are possible, but this is the most usual ones. If you have other exposition than LDAP and SQL, this is also possible to bind to it.

You have all the documentation about the User Management configuration into the User Management page.

If the authentication against your infrastructure is particular - you have an SSO system, or others - the Nuxeo Platform has extension points that will let you adapt your Nuxeo application to your infrastructure. The documentation about that is the Authentication section.

Creating a Nuxeo Static WAR

Creating a static WAR of Nuxeo allows you to deploy Nuxeo in a setting where your Tomcat application server only allows WAR installation. Keep in mind that when you do this, the following dynamic features will **not** work (we are inside a **static** war):

- Nuxeo Studio hot reload
- Nuxeo IDE hot reload
- Nuxeo Marketplace integration (Hotfixes and packages installation)

Creating a Nuxeo Static WAR Distribution

Here is the way to create your own static WAR distribution:

1. Download a Nuxeo Tomcat distribution.
2. Unzip it in somewhere (let's call it `$NUXEO_HOME`).
3. Copy your specific bundles into `$NUXEO_HOME/nxserver/bundles/`.
4. Create your own templates and configuration files as needed.
5. Start your Nuxeo Tomcat instance as usual .
6. Check that your Nuxeo is well configured (database configuration, ldap configuration, etc...).
7. Stop your Nuxeo instance.
8. Launch this following command:

```
$NUXEO_HOME/bin/nuxeoctl pack /tmp/nuxeo-war.zip
```

Your static WAR distribution will be generated into `/tmp/nuxeo-war.zip`.

The generated ZIP contains what needs to be copied to your Tomcat installation:

```
nuxeo-war.zip/
|-- endorsed (jaxb and jaxws api libs that should replace packages provided by default
JDK that are outdated)
|-- lib (nuxeo common libs)
|-- README-NUXEO.txt
`-- webapps
    `-- nuxeo (exploded WAR that you can zip if needed)
```

Installing Nuxeo in the Target Tomcat

The `README-NUXEO.txt` gives you the needed instructions. Here is an example of this file (the JDBC driver configuration, username and password will vary depending on the Nuxeo configuration choices you made before running `nuxeoctl pack`):

```
This ZIP must be uncompressed at the root of your Tomcat instance.

In order for Nuxeo to run, the following Resource defining your JDBC datasource
configuration
must be added inside the <GlobalNamingResources> section of the file conf/server.xml

  <Resource auth="Container" driverClassName="org.h2.Driver" maxActive="100"
maxIdle="30" maxWait="10000" name="jdbc/nuxeo" password="" type="javax.sql.DataSource"
url="jdbc:h2:nuxeo" username="sa" validationQuery=""/>

Make sure that the 'url' attribute above is correct.
Note that the following file also contains database configuration:

  webapps/nuxeo/WEB-INF/default-repository-config.xml

Also note that you should start Tomcat with more memory than its default, for
instance:

  JAVA_OPTS="-Xms512m -Xmx1024m -XX:MaxPermSize=512m -Dnuxeo.log.dir=logs"
bin/catalina.sh start
```

So basically what you need to do is:

1. Copy the structure (endorsed, lib, webapp) inside your target Tomcat.
2. Copy the `<Resource>` tag as described in the `README-NUXEO.txt` into your `server.xml` in order to declare the JDBC datasources.

The target database will be the one you defined in your source Nuxeo instance, and the default for the Nuxeo directories are:

- `nuxeo.config.dir` : `WEB-INF` directory,
- `nuxeo.runtime.home` : `$TOMCAT_HOME/nuxeo`,
- `nuxeo.data.dir` : `$TOMCAT_HOME/nuxeo/data`,
- `nuxeo.tmp.dir` : `$TOMCAT_HOME/nuxeo/tmp`,
- `nuxeo.web.dir` : `$TOMCAT_HOME/nuxeo/web` (for WebEngine modules).

See Configuration Parameters Index (nuxeo.conf) for more details about these config parameters of Nuxeo.

If you need to change the values for these paths, you can add parameters in the file `webapps/nuxeo/WEB-INF/web.xml`. See JavaDoc of NuxeoStarter for more details.

Firewall Consideration

Firewalls don't like inactive connection that stay open. Most of them will drop the connection silently, which will generate errors on connections pools like database, AJP or LDAP. Here are some advices to prevent this.

Firewall Between Apache and Nuxeo in AJP Mode

If you are using mod_proxy_ajp, you can activate a TCP keep alive to prevent persistent connections to be dropped. This requires the usage of mod_proxy options "keepalive=on" and "flushpackets=on". You also need to configure the TCP keep alive delay using sysctl (`net.ipv4.TCP_keepalive_time`).

Refer to mod_proxy documentation for more information.

Firewall Between Nuxeo and the Database

Since Nuxeo Platform 5.5.0, database pool can try to reconnect on invalid connection (NXP-7528) but it is better to enable the keep alive on your database. For instance using PostgreSQL, this can be achieved with the following options:

Option	Value	Description
`tcp_keepalives_count`	5	Maximum number of TCP keepalive retransmits.
`tcp_keepalives_idle`	60	Time between issuing TCP keepalives.
`tcp_keepalives_interval`	60	Time between TCP keepalive retransmits.

Firewall Between Nuxeo and LDAP

Here, there are no keep alive alternative. You can simply disable the LDAP connection pool in the directory configuration.

JDBC Datasource Configuration

Datasource Definition

Nuxeo code and Nuxeo configuration uses JDBC connections for a number of purposes, and therefore defines different datasource names for them. Examples are:

- `jdbc/nxsqldirectory`
- `jdbc/nxaudit-logs`
- `jdbc/NuxeoDS`

Additional datasources can be used, for instance when defining a Directory you have to specify which datasource it uses, and you may want to use something else than the standard `jdbc/nxsqldirectory` if you want to store data elsewhere.

In this section

These datasources are all defined in Tomcat's `conf/Catalina/localhost/nuxeo.xml` (which is generated from `templates/common-base/conf/Catalina/localhost/nuxeo.xml.nxftl` — to defined new datasource names you should copy this template and override it). The `nuxeo.xml` datasources are defined like this:

```
<ResourceLink name="jdbc/NuxeoDS" global="jdbc/nuxeo" type="javax.sql.DataSource" />
<ResourceLink name="jdbc/nxsqldirectory" global="jdbc/nuxeo"
type="javax.sql.DataSource" />
...
```

So by default they are actually links to a global resource defined in Tomcat's `conf/server.xml` (generated from `templates/common-base/conf/server.xml.nxftl`):

```
<Resource name="jdbc/nuxeo" auth="Container" type="javax.sql.DataSource"
    maxActive="${nuxeo.db["max-pool-size"]}" maxIdle="30" maxWait="10000"
driverClassName="${nuxeo.db.driver}"
    url="${nuxeo.db.jdbc.url}" validationQuery="${nuxeo.db.validationQuery}"
    username="${nuxeo.db.user}" password="${nuxeo.db.password}"
    accessToUnderlyingConnectionAllowed="true" />
```

The properties used in this file come from the ones defined in `bin/nuxeo.conf` and the template chosen for your database.

Single-Datasource Mode (non-XA)

By default since Nuxeo 5.7.1 a mode called *single-datasource* is activated (see NXP-10308 for technical details).

In this mode, every database connection made by Nuxeo is funneled through a single physical datasource, including:

- Directory connections,
- VCS connections,
- Hibernate connections.

Even if you defined a different datasource name for a directory, with this mode activated a single shared datasource will actually be used instead. This allows the use of a regular datasource for everything, and avoids the use of XA, which is a big performance boost for some databases (like Oracle in RAC mode).

Single-datasource mode applies even to VCS, which means that in this mode the `<xa-datasource>` configuration and the various database `<property>` defined by `nxserver/config/default-repository-config.xml` will not be used.

To activate single-datasource mode, the following is defined in `templates/common-base/nuxeo.defaults`:

```
nuxeo.db.singleDataSource=jdbc/NuxeoDS
```

The datasource name can of course be different than `jdbc/NuxeoDS`, but it must be a valid datasource already defined elsewhere.

In addition, the maximum pool size for the datasource must be enough for all the VCS connections, which means that you must have `nuxeo.db.max-pool-size` `nuxeo.vcs.max-pool-size` + 2. (The + 2 comes from two reserved connections for the VCS lock manager and the cluster node handler.)

As an advanced feature, if you want single-datasource mode *except* for some specific datasources (that need to hit a separate database for instance), then you can use:

```
nuxeo.db.singleDataSource.exclude=jdbc/myExcludedDatasource,
jdbc/anotherExcludedDatasource
```

The excluded datasources will not participate in any global XA transaction so there will be no two-phase commit for them, and on error and rollback (which admittedly should not happen) they may become inconsistent.

Disabling Single-Datasource Mode

You will have to disable single-datasource mode in these situations:

- You have more than one SQL repository,
- You want to go back to the old XA mechanism for multiple datasource.

If either of this is the case, then you will have to disable single-datasource mode, by redefining the property in `bin/nuxeo.conf` with an empty value:

```
nuxeo.db.singleDataSource=
```

Using Single-Datasource Mode in Java Code

If you code a new Java component that needs to access JDBC directly and wish to use single-datasource mode, then you should get your connections through:

```
// try single-datasource mode first
        Connection connection = ConnectionHelper.getConnection(dataSourceName);
        try {
            if (connection == null) {
                // standard mode, get the connection through regular JDBC lookup
                connection =
DataSourceHelper.getDataSource(dataSourceName).getConnection();
            }
            ... use the connection ...
        } finally {
            if (connection != null) {
                connection.close();
            }
        }
```

Nuxeo Clustering Configuration

Nuxeo can be clustered between several nodes (a.k.a. instances or machines) with the appropriate configuration. In addition, a HTTP load balancer with session affinity must be used in front of the nodes.

Requirements

To enable clustering, you must have at least two nodes with:

- a shared database,
- a shared filesystem (unless you use an external binary store like S3),
- a load-balancer with stick sessions.

The shared filesystem is usually a NFS mount. You **must not** shared the whole Nuxeo installation tree, see below.

The load balancer **must** use sticky sessions if the clustering delay is not 0. Having a non-0 clustering delay is recommended for performance reasons. See below for more.

In this section

Shared Filesystem Configuration

The complete Nuxeo instance hierarchy **must not** be shared between all instances. However a few things must or should be shared.

Binaries

The `repository.binary.store` (`nxserver/data/binaries` by default) directory **must** be shared by all Nuxeo instances in order for VCS to function correctly.

Temporary Directory

The temporary directory configured through `nuxeo.tmp.dir` **must not** be shared by all instances, because there are still a few name collision issues that may occur, especially during startup.

However, in order for various no-copy optimizations to be effective, the temporary directory should be on the same filesystem as the binaries directory. To do this, the recommended way is to have each instance's `nuxeo.tmp.dir` point to a different subdirectory of the shared filesystem.

VCS Cluster Configuration

Setup

To set up clustering, please update the `repository.clustering.enabled`, `repository.clustering.delay` and `repository.binary.store` in the `nuxeo.conf` parameters. On all Nuxeo instances, the `repository.binary.store` should point to a shared filesystem unless you use an external binary store like S3.

The cluster nodes must only share the `binaries` folder (configured with `repository.binary.store`), not the entire data directory (configured with `nuxeo.data.dir`): the reason is the data directory contains data related to features that are not working in a cluster environment, in particular everything related to the Marketplace package management.

The cluster setup parameters are:

- **`repository.clustering.enabled`** must be `true` to enable clustering.
- **`repository.clustering.delay`** is expressed in milliseconds, and specifies a delay during which invalidations don't need to be processed. Using a non-0 value is an important optimization as otherwise every single transaction, even a read-only one, would have to hit the database to check invalidations between several nodes. However this means that one node may not see immediately the changes made on another node, which is a problem if you don't use sticky session on the load balancer.
- **`repository.binary.store`** must point to a shared storage. Under Windows, the path value can be UNC formatted, for instance `\\servername\sharename`.
- **`nuxeo.db.validationQuery`** must contain a SELECT clause for validating connections in the pool according to your database type. For instance `SELECT 1` used on PostgreSQL or `SELECT 1 FROM dual` on Oracle.

There is a dedicated page detailing all the VCS configuration options.

Checking the SQL Tables Initialization

Start the SQL server, all Nuxeo nodes (the first alone and the other afterwards to avoid concurrent initialization of the SQL tables) and the load balancer and login on the HTTP user interface on each cluster node, then check that on the database that the `cluster_nodes` table is initialized with one line per node:

```
nuxeo-db=# select * from cluster_nodes;
 nodeid |          created
--------+---------------------------
  25767 | 2009-07-29 14:36:08.769657
  32546 | 2009-07-29 14:39:18.437264
(2 lines)
```

Checking VCS Cache Invalidations

Create a document and browse it from two different nodes. Edit the title from one node and navigate back to the document from second node to check that the change is visible. You can also monitor what's happening in the `cluster_invals` table to see cache invalidation information.

Quartz Scheduler Cluster Configuration

The Quartz scheduler should be configured to run in a cluster. This is needed in order for scheduled events, like periodic cleanups or periodic imports, to be executed only on one node and not on all nodes at the same time, which could cause problems.

Standard configuration is available from Nuxeo templates for Tomcat for PostgreSQL, Oracle and SQL Server.

First, you should populate the database with the tables needed by quartz (names `QRTZ_*`). The DDL scripts come from the standard Quartz distribution and are available in the Nuxeo templates in `$NUXEO_HOME/templates/<database>-quartz-cluster/bin/create-quartz-tables.sql`.

Second, you should enable the quartz-specific cluster templates by adding the template `<database>-quartz-cluster`.

In cluster mode the schedule contributions (deployed from plugins or configuration files) **must** be the same on all nodes.

HTTP Load Balancer Configuration

Set up an HTTP or AJP load balancer such as Apache with `mod_proxy or mod_proxy_ajp` or Pound, and configure it to keep session affinity by tracking the value of the `JSESSIONID` cookie and the "`;jsessionid`" URL parameter.

If you use a stateless load balancer such as apache modules such as `mod_jk` and `mod_proxy_balancer`, you need to make the HTTP server generate `JSESSIONID` cookies with values that end with `.nxworkern`, where `nxworkern` is a string suffix specific to each node (you can use any string).

To do so, in `nuxeo.conf` specify a different `nuxeo.server.jvmRoute` for each node, for instance `nuxeo.server.jvmRoute=nxworker1`. This will instruct the Nuxeo preprocessing phase to correctly fill the `jvmRoute` attribute of the `Engine` element in the generated `server.xml`.

Then configure you stateless balancer to follow these routes, for instance here is the relevant configuration fragment when using `mod_proxy_balancer`:

```
ProxyPass /nuxeo balancer://sticky-balancer stickysession=JSESSIONID|jsessionid
nofailover=On

<Proxy balancer://sticky-balancer>
  BalancerMember http://192.168.2.101:8080/nuxeo route=nxworker1
  BalancerMember http://192.168.2.102:8080/nuxeo route=nxworker2
</Proxy>
```

Troubleshooting Session Affinity Problems

To test that the load balancer forwards the HTTP requests of a given session to the same node you can add a new file on each node (after Tomcat started), `$NUXEO_HOME/nxserver/nuxeo.war/clusterinfo.html`, with on the first node:

```
<html><body>Node 1</body></html>
```

and on the second node:

```
<html><body>Node 2</body></html>
```

Using a browser with an active Nuxeo session (an already logged-in user), then go to `http://yourloadbalancer/nuxeo/clusterinfo.html` and check that you always return to the same node when hitting the refresh button of the browser.

Related pages

- Examples of SQL Generated by VCS (Nuxeo Platform Developer Documentation - 5.8)
- Internal VCS Model (Nuxeo Platform Developer Documentation - 5.8)
- Java Data Structures and Caching (Nuxeo Platform Developer Documentation - 5.8)
- Nuxeo Clustering Configuration (Nuxeo Installation and Administration - 5.8)
- Performance Recommendations (Nuxeo Platform Developer Documentation - 5.8)
- Redis Configuration (Nuxeo Installation and Administration - 5.8)
- Setting up a HA Configuration Using the Nuxeo Platform and PostgreSQL (Nuxeo Installation and Administration - 5.8)
- VCS Architecture (Nuxeo Platform Developer Documentation - 5.8)
- VCS Configuration (Nuxeo Installation and Administration - 5.8)

VCS Tables (Nuxeo Platform Developer Documentation - 5.8)

Redis Configuration

Nuxeo instances should be configured with a Redis server (in addition to the regular SQL database) whenever:

- It's important that asynchronous jobs not yet executed be kept across server restarts.
- In cluster mode and some asynchronous jobs should be executed only on some nodes (for instance image conversion or fulltext extraction).

For a robust production instance, the first point is always necessary, which means that Redis should always be used.

Configuring Redis

Redis 2.6 or higher must be installed preferably on a separate server.

The following Redis configuration points should be checked:

- The server memory should be enough to hold the Redis database (which is not expected to be big in Nuxeo Platform 5.8 unless there is a huge backlog of asynchronous jobs).
- Redis persistence should be configured appropriately for the level of service required. In particular the RDB files should be used as backups and periodically saved offsite.
- Redis master-slave replication should be set up, for robustness (fast disaster recovery). Note that Nuxeo Platform 5.8 does not yet know how to use the slaves for read-only operation.

Configuring Nuxeo for Redis

To make the Nuxeo Platform use Redis, you must activate the following in `bin/nuxeo.conf`:

```
nuxeo.redis.enabled=true
nuxeo.redis.host=redishost
```

The `nuxeo.redis.host` must be the hostname or IP address of your master Redis server. All the Nuxeo instances in a Nuxeo cluster must of course point to the same Redis server.

Also available are (with defaults):

```
nuxeo.redis.port=6379
nuxeo.redis.prefix=nuxeo:work:
nuxeo.redis.password=
nuxeo.redis.database=0
nuxeo.redis.timeout=2000
```

The `nuxeo.redis.port` is self-explanatory, 6379 is the value for a default Redis installation.

The `nuxeo.redis.prefix` is the prefix used for all Nuxeo keys stored and read in Redis. This allows you to use a single Redis server between several Nuxeo cluster installations by having a different prefix for each cluster, but this is not really recommended.

The `nuxeo.redis.password`, `nuxeo.redis.database` and `nuxeo.redis.timeout` are standard Redis parameters, although rarely used.

When `nuxeo.redis.enabled=true` then the following is automatically activated as well:

```
nuxeo.work.queuing=redis
```

(As of Nuxeo Platform 5.8, work queuing is the only use of Redis in the standard Nuxeo modules, so it makes sense to activate both together.)

VCS Configuration

VCS (Visible Content Store) is the default storage engine for Nuxeo documents.

The following are the options available to configure VCS repository in Nuxeo Platform. They usually go in a file named `default-repository-config.xml`.

In a standard Nuxeo this file is generated from a template, and many elements or attributes actually take their values from parameters in nuxeo.conf.

Example File

This file is for illustration and contains many more options than are necessary by default.

```
<?xml version="1.0"?>
<component name="default-repository-config">
  <extension target="org.nuxeo.ecm.core.repository.RepositoryService"
point="repository">
    <repository name="default"
      factory="org.nuxeo.ecm.core.storage.sql.coremodel.SQLRepositoryFactory">
      <repository name="default">
        <pool minPoolSize="0" maxPoolSize="20"
          blockingTimeoutMillis="100" idleTimeoutMinutes="10" />
        <clustering enabled="true" delay="1000" />
        <idType>varchar</idType>
        <schema>
          <field type="largetext">note</field>
        </schema>
        <indexing>
          <includedTypes>
            <type>File</type>
            <type>Note</type>
          </includedTypes>
          <!-- sample for excluded types -->
          <!--
          <excludedTypes>
            <type>Root</type>
            <type>Workspace</type>
          </excludedTypes>
          -->
          <fulltext analyzer="english"> <!-- PostgreSQL -->
            <index name="default">
              <!-- all props implied -->
            </index>
            <index name="title">
              <field>dc:title</field>
            </index>
            <index name="description">
              <field>dc:description</field>
              <excludeField>content/data</excludeField>
            </index>
          </fulltext>
          <queryMaker class="org.nuxeo.ecm.core.storage.sql.NXQLQueryMaker" />
          <queryMaker class="org.nuxeo.ecm.core.chemistry.impl.CMISQLQueryMaker" />
        </indexing>
        <binaryStore path="binaries"/>
        <binaryManager class="org.nuxeo.ecm.core.storage.sql.DefaultBinaryManager"/>
        <usersSeparator key="," />
        <aclOptimizations enabled="true" readAclMaxSize="4096"/>
        <pathOptimizations enabled="true"/>
        <noDDL>false</noDDL>
        <sqlInitFile>myconf.sql.txt</sqlInitFile>
      </repository>
    </repository>
  </extension>
</component>
```

Pooling Options

```
<pool minPoolSize="0" maxPoolSize="20"
  blockingTimeoutMillis="100" idleTimeoutMinutes="10" />
```

- minPoolSize: the minimum pool size (default is **0**) (see `nuxeo.vcs.min-pool-size` in `nuxeo.conf`).
- maxPoolSize: the maximum pool size, above which connections will be refused (default is **20**) (see `nuxeo.vcs.max-pool-size` in `nuxeo.conf`).
- blockingTimeoutMillis: the maximum time (in milliseconds) the pool will wait for a new connection to be available before deciding that it cannot answer a connection request (pool saturated).
- idleTimeoutMinutes: the time (in minutes) after which an unused pool connection will be destroyed.

This is available since Nuxeo 5.6 (see NXP-9763), only when using using Tomcat. (Before Nuxeo 5.6, the pool was configured through `NuxeoConnectionManager <Resource>` in the `nuxeo.xml`; when using JBoss, the pool is configured through `default-repository-ds.xml`).

Clustering Options

```
<clustering enabled="true" delay="1000" />
```

- clustering enabled: use **true** to activate Nuxeo clustering (default is **false**, i.e., no clustering) (see `repository.clustering.enabled` in `nuxeo.conf`).
- clustering delay: a configurable delay in milliseconds between two checks at the start of each transaction, to know if there are any remote invalidations (see `repository.clustering.delay` in `nuxeo.conf`).

Column Types

Large Text / CLOB Columns

```
<schema>
  <field type="largetext">note</field>
  <field type="largetext">my:field</field>
  ...
</schema>
```

- field type="largetext": a field that should be stored as a CLOB column inside the database instead of a standard VARCHAR column.

This is important for your large text fields, especially for MySQL, Oracle and SQL Server which have very small defaults for standard text fields.

Using Oracle, if you attempt to save a string too big for the standard NVARCHAR2(2000) field, you will get the error:

```
java.sql.SQLException: ORA-01461: can bind a LONG value only for insert into a LONG
column
```

Note that since Nuxeo 5.4.2 using this method is not recommended anymore, you should instead use restrictions in the XML Schemas for your type to specify length constraints (see NXP-6301).

Id Column Type

In standard Nuxeo the document id is a UUID stored as a string, for instance "`9ea9a461-e131-4127-9a57-08b5b9b80ecb`".

Starting with Nuxeo 5.7.1, it's possible on select databases to use a more efficient id representation:

```
 <idType>varchar</idType>
```

The following values for idType are possible:

- **varchar**: a varchar-based UUID (the default),

- **uuid**: a native uuid (only on PostgreSQL (NXP-4803)),
- **sequence**: a sequence-based integer (on PostgreSQL (NXP-10894) and SQL Server 2012 (not Azure) (NXP-10912)). Instead of just `sequence` you can also use `sequence:your_sequence_name` if you want to use another sequence than the default one (`hierarchy_seq`).

When using a sequence, the document ids will be simple incremental small integers instead of randomly-generated UUIDs.

Note that switching this option to a new value will require a full dump, manual conversion and restore of your database, so it should be specified before starting Nuxeo for the first time.

Indexing Options

Configuring Which Types Will Be Indexed

Since Nuxeo DM 5.5 it is possible to configure the document types you want to index or you want to exclude from fulltext indexing. This is possible using the tags `includedTypes` and `excludedTypes` inside the `indexing` tag:

```
<includedTypes>
  <type>File</type>
  <type>Note</type>
</includedTypes>
```

or

```
<excludedTypes>
  <type>Root</type>
  <type>Workspace</type>
</excludedTypes>
```

If you set both included and excluded types, only the included types configuration will be taken into account.

Fulltext

```
<fulltext disabled="true" analyzer="english" catalog="...">
  ...
</fulltext>
```

- fulltext disabled: use **true** to disable fulltext support, the repository configuration must be updated to have (default is **false**, i.e., fulltext enabled).
- fulltext analyzer: a fulltext analyzer, the content of this attribute depends on the backend used:
  - H2: a Lucene analyzer, for instance `org.apache.lucene.analysis.fr.FrenchAnalyzer`. The default is an english analyzer.
  - PostgreSQL: a Text Search configuration, for instance `french`. The default is **english**. See http://www.postgresql.org/docs/8.3/static/textsearch-configuration.html
  - Oracle: an Oracle PARAMETERS for fulltext, as defined by http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/cdatadic.htm (see NXP-4035 for details).
  - Microsoft SQL Server: a fulltext LANGUAGE, for instance `english`, as defined in http://msdn.microsoft.com/en-us/library/ms187787(v=SQL.90).aspx. The default is **english**.
  - other backends don't have configurable fulltext analyzers.
- fulltext catalog: a fulltext catalog, the content of this attribute depends on the backend used:
  - Microsoft SQL Server: a fulltext CATALOG, the default is **nuxeo**.
  - other backends don't need a catalog.

Fulltext indexes are queried in NXQL through the `ecm:fulltext` pseudo-field. A non-default index "foo" can be queried using `ecm:fulltext_foo`.

If no `<index>` elements are present, then a **default** index with all string and blob fields is used.

```
<fulltext ...>
  <index name="title" analyzer="..." catalog="...">
    <field>dc:title</field>
    <field>dc:description</field>
  </index>
  <index name="blobs">
    <fieldType>blob</fieldType>
  </index>
  <index name="other">
    <fieldType>string</fieldType>
    <excludeField>dc:title</excludeField >
  </index>
</fulltext>
```

- index name: the name of the index (the default is **default**).
- index analyzer: a fulltext analyzer just for this index. See fulltext options above for details.
- index catalog: a fulltext catalog just for this index. See fulltext options above for details.
- fieldType: **string** or **blob**, the default being both. This selects all these fields for indexing.
- field: the name of a field that should be selected for indexing.
- excludeField: the name of a field that should not be in the index.

If no `<fieldType>`, `<field>` or `<excludeField>` is present, then all string and blob fields are used.

Query Maker

```
<queryMaker class="..."/>
```

- queryMaker class: registers a `QueryMaker`, to extend the query system. The class must implement `org.nuxeo.ecm.core.storage.sql.QueryMaker` (the default is `org.nuxeo.ecm.core.storage.sql.NXQLQueryMaker`, i.e., the standard NXQL QueryMaker).

There can be serveral `<queryMaker>` elements defined.

This is not needed (and deprecated) starting with Nuxeo EP 5.3.2.

Binary Store

```
<binaryManager class="org.nuxeo.ecm.core.storage.sql.XORBinaryManager" key="abc"/>
```

- binaryManager class: the default Binary Manager can be changed using this (the default is to use the standard binary manager that stores files normally). A new XORBinaryManager has been added, it knows how to do XOR with a pattern on read/write (see the key below). The on-disk binary store is unchanged (the hash of the files is still the filename), but of course it's now unreadable by humans by default. One consequence is that for the same file the application-level digest in the Binary object is now different if encryption is enabled.
- binaryManager key: the encryption key for the binary manager (if it's doing any encryption). Changing this value will of course render existing binaries unreadable.

```
<binaryStore path="/foo/bar"/>
```

- binaryStore path: the filesystem path where the binary store should live. A relative path is interpreted relative to the Nuxeo Framework home. The default is the **binaries** directory. (See `repository.binary.store` in `nuxeo.conf`.)

Optimizations

```
<pathOptimizations enabled="false"/>
```

- pathOptimizations enabled: for PostgreSQL, Oracle and MS SQL Server (and H2), it is possible to disable the path-based optimizations by using **false** (the default is **true**, i.e., path optimizations enabled).

```
<aclOptimizations enabled="false"/>
```

- aclOptimizations enabled: for PostgreSQL, Oracle and MS SQL Server (and H2), you can disable the read ACL optimizations by using **false** (the default is **true**, i.e., ACL optimizations enabled).
- since DM 5.4.1, you can set the property readAclMaxSize to define the size of the larger ACL for a document : this may be useful if you have mainly affected permissions to a lot of users, instead of using groups (do not set this attribute if you disable ACL optimizations).

```
<usersSeparator key="," />
```

- in case the user/group names in your directories contains the separator character used in the Read ACL cache(comma by default), you can change this value using the attribute `usersSeparator`
- if you change this value on an existing database, you will need to rebuild the ACL cache with the SQL command:
SELECT nx_rebuild_read_acls();

Database Creation Option

```
<noDDL>true</noDDL>
```

Set the value noDDL to **true** to execute no DDL (Data Definition Language). The default is **false**.

When this is **true**, VCS will assume that no new structure has to be created in the database. This means that none of these statements will be executed:

- `CREATE TABLE`, `CREATE INDEX`, `ALTER TABLE ADD CONSTRAINT` for a new schema or complex property,
- `ALTER TABLE ADD column` for a new property in a schema,
- `CREATE FUNCTION`, `CREATE PROCEDURE`, `CREATE TRIGGER` for VCS internal stored procedures and migration steps.

The only statements that VCS will execute are:

- `INSERT`, `UPDATE`, `DELETE` for data changes,
- calling of stored procedures.

This means that all tables, indexes, triggers and stored procedures needed by VCS have to be created beforehand, either by a previous execution when the flag was **false**, or by a manual execution of a SQL script from a previously-created Nuxeo instance.

This option is typically needed if you configure the VCS connection with a username who is not the owner of the database, usually for security considerations.

```
<sqlInitFile>myconf.sql.txt</sqlInitFile>
```

If you need to execute additional SQL when the database is initialized (at every Nuxeo startup), you can use this to specify an additional SQL file to read and execute (unless noDDL is true). The format of an SQL init file is described below. Examples can be found in the standard SQL init files used by Nuxeo, which are available at https://github.com/nuxeo/nuxeo-core/tree/release-5.8/nuxeo-core-storage-sql/nuxeo-core-storage-sql/src/main/resources/nuxeovcs (in the appropriate branch for your version).

A SQL init file is a series of SQL statements.

A # starting a line (as the first character) makes the line a comment (ignored), except for a few special #-starting tags (see below).

SQL statements have to be separated from every other by a **blank line**.

A statement may be preceded by one or more **tag** lines, which are lines starting with **`#SOMETAG:`** (including the final colon), and may be:

- `#CATEGORY:` defines the category for all future statements, until a new category is defined. See below for the use of categories.
- `#TEST:` specifies that the following statement returns a certain number of rows, and if that number of rows is 0 then the variable `emptyResult` will be set to true, otherwise it will be set to false.
- `#IF: variable` or `#IF: ! variable`, conditions execution of the single following statement on the value of the **variable**. Several `#IF:` tags may be used in a row (in different lines), and they are effectively ANDed together.

The following boolean variables are predefined by Nuxeo and the various database dialects, and may be use in `#IF:` tags:

- `emptyResult`: true if the previous `#TEST:` statement returned no row,
- `fulltextEnabled`: true if fulltext is enabled in the repository configuration,
- `clusteringEnabled`: true if clustering is enabled,
- `aclOptimizationsEnabled`: true if ACL optimizations are enabled,
- `pathOptimizationsEnabled`: true if path optimizations are enabled,
- `proxiesEnabled`: true if proxies are enabled,
- `softDeleteEnabled`: true if soft delete is enabled,
- `sequenceEnabled`: true if sequence-based ids are enabled.

Note that not all dialects define all variables, consult the specific dialect code or the standard Nuxeo SQL init file to know more.

SQL statements are regular SQL statements and will be executed as-is by the database, with variable substitution (see below). Depending on the dialect, it may or may not be necessary of forbidden to end some kinds of statement with a semicolon, please consult the standard Nuxeo SQL init file for the dialect to be sure. Note also that when writing multi-line stored procedures, you must not include a blank line for readability, as this blank line would be interpreted as the end of the whole multi-line SQL statement.

The following variables provide additional dialect-specific values that may be used in SQL statements using the variable substitution syntax **`${variablename}`**:

- `idType`: the SQL type used for ids,
- `idTypeParam`: the SQL type used for ids in stored procedures (not all dialects use this),
- `idSequenceName`: when sequence-based ids are enabled, the name of the sequence to use,
- `idNotPresent`: a representation of a "marker" id to use in stored procedures to represent a non-existent id,
- `fulltextAnalyzer`: the fulltext analyzer defined in the repository configuration.

A few pseudo-SQL statements can be used to provide addition logging actions:

- `LOG.DEBUG message`: logs the message at DEBUG level in the standard logger,
- `LOG.INFO message`: logs the message at INFO level in the standard logger,
- `LOG.ERROR message`: logs the message at ERROR level in the standard logger,
- `LOG.FATAL message`: logs the message at ERROR level in the standard logger and throws an exception that will stop database intialization and make it unusable by Nuxeo.

To initialize the database, the statements of the following categories are executed in this order:

- `first`
- `beforeTableCreation`
- (at this point Nuxeo does a CREATE or ALTER on the tables based on the Nuxeo Schema definitions)
- `afterTableCreation`
- `last`

The following categories are executed in special circumstances:

- `addClusterNode`: when creating a cluster node,
- `removeClusterNode`: when shutting down a cluster node.

Enabling Drag and Drop for Internet Explorer

Since Tomcat 7, drag and drop doesn't work on Internet Explorer because of a security option enabled by default. Indeed Tomcat prevents cookies access from JavaScript (see NXP-12202 for details). Here is a workaround to enable drag and drop on this browser.

1. Make sure the server is stopped.
2. Edit the file `$TOMCAT/conf/context.xml`: add `useHttpOnly="false"` on Context element.
3. Start the server.

   Users can now use the Internet Explorer drag and drop extension.

Other documentation about drag and drop

- Enabling Drag and Drop for Internet Explorer (Nuxeo Installation and Administration - 5.8)
- Drag and Drop Compatibility Table (Nuxeo Platform User Documentation - 5.8)
- Working Using Drag and Drop (Nuxeo Platform User Documentation - 5.8)
- Importing Content Using Drag and Drop (Nuxeo Platform User Documentation - 5.8)
- Installing Drag and Drop Extensions (Nuxeo Platform User Documentation - 5.8)
- Drag and Drop Service for Content Capture (HTML5-Based) (Nuxeo Platform Developer Documentation - 5.8)
- Importing Assets in DAM (Nuxeo Platform User Documentation - 5.8)

Read ACLs

ACL Management Modes

Two modes are available to configure the management of ACLs in the Nuxeo Platform. They are activated using the `aclOptimization` parameter in the repository configuration.

- ReadACL OFF (`aclOptimization` to false) only writes the permissions in the ACL table: this mode is costless at the write time (document creation, permissions changes) but has an important cost at read time, when filtering the results of a search.
- ReadACL ON (`aclOptimization` to true) is the default one in the Nuxeo Platform. It computes the READ ACLs for every document at document creation or permission settings: not only permissions are written in the ACL table, but other tables are updated to provide an efficient cache when launching queries. So this mode is fast at read time, but may be expensive at write time.

ReadACL ON is the default choice because in most cases people would prefer to promote better search performances rather that document update performances. Quick search results are usually considered as more critical.

In this section

Potential Issues

A technical consequence of this trade-off is a case where the Nuxeo Platform may encounter problems: setting new permission on a big folder (like a domain) may result in a timeout when saving the changes in the **Access rights** tab.

Possible solutions

The main solution is to:

1. Use groups dedicated to these big folders.
2. Add users and subgroups to these groups instead of changing the permissions on these big folders.

Another solution would be to save the permissions asynchonously: A persisted job needs to be scheduled for this task, to provide the opportunity to resume the work if the Nuxeo Platform is stopped (shutdown or unexpected failure).

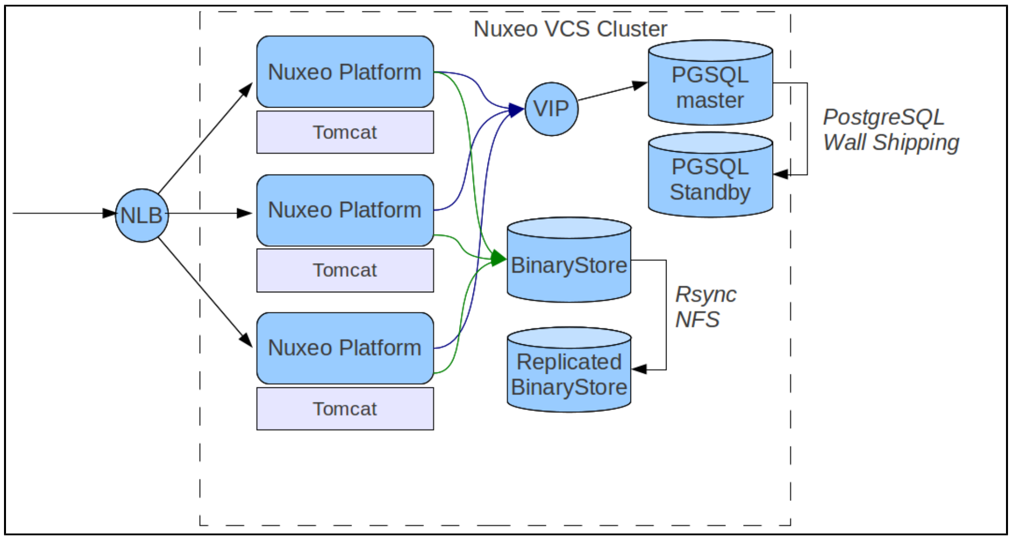
Nuxeo will work on this solution for the next release, now that persisted jobs are available in the framework. However, this will not prevent to have an important activity on the database side, that would impact the overall performance of the application, during this update.

Setting up a HA Configuration Using the Nuxeo Platform and PostgreSQL

Target Architecture

The target architecture is to:

- Use the Nuxeo Platform built-in clustering mode to ensure HA at Application level;
- Configure PostgreSQL Master/Slave replication mode to ensure HA at Database level.



On this page

Nuxeo Clustering and Network Load-Balancing

See the Nuxeo Clustering Configuration page.

Setting up PosgreSQL Streaming Replication

Streaming replication allows a standby server to stay more up-to-date than is possible with file-based log shipping.

We provide ansible scripts to deploy a cluster of two PostgreSQL servers as an example. Please refer to the PostgreSQL streaming replication page for more information.

> There are several ways to achieve PGSQL replication: we are just presenting one of the possible option.

BinaryStore Replication

Technically, you don't need to replicate the BinaryStore if you use a reliable backend like:

- a NAS that already handles redundancy;
- Amazon S3.

However, if you want to have the full storage replicated in two separated data center, you will want to replicate the BinaryStore too.

Because of the way the BinaryStore is handled (no update, move) , you don't have a lot of constraints:

- Rsync
- DRBD
- ...

PostgreSQL Fail over Procedure

When the master database is down, the slave must be promoted to master. This means changing changing the `recovery.conf` and `postgresql.conf` files and restart. This can be easily scripted.

If your infrastructure provides a virtual IP for the database, the VCS pool (used for the document repository access) is able to reconnect automatically to the database (NXP-7528). In addition you need to a define a `validationQuery` for the default db datasource (used for directory or audit access).

Backup and Restore

Please refer to the Nuxeo backup procedure.

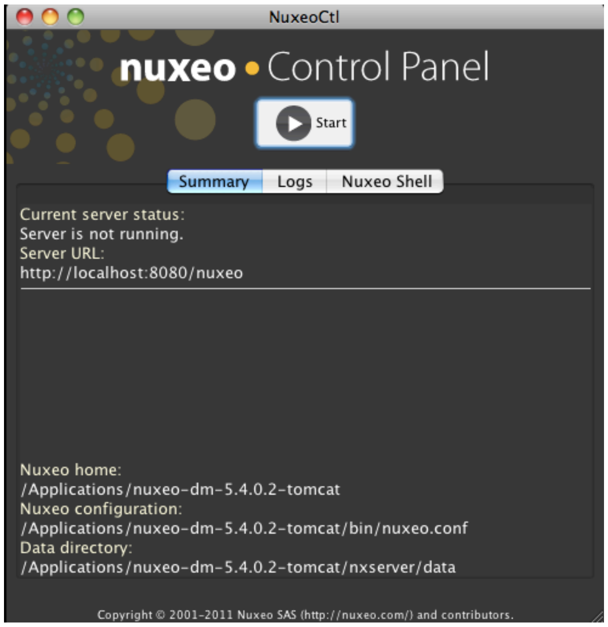
Related pages

- Setting up a HA Configuration Using the Nuxeo Platform and PostgreSQL (Nuxeo Installation and Administration - 5.8)
- How to Replicate the Nuxeo Repository (Nuxeo Installation and Administration - 5.8)
- Redis Configuration (Nuxeo Installation and Administration - 5.8)
- Nuxeo Clustering Configuration (Nuxeo Installation and Administration - 5.8)
- VCS Tables (Nuxeo Platform Developer Documentation - 5.8)

Server Start and Stop

On this page, you will see how to start and stop your Nuxeo application.

Nuxeo applications come with a Control Panel that allows you to start and stop the server easily, and to access more administration features.



```
Current server status:
Server is not running.
Server URL:
http://localhost:8080/nuxeo

Nuxeo home:
/Applications/nuxeo-dm-5.4.0.2-tomcat
Nuxeo configuration:
/Applications/nuxeo-dm-5.4.0.2-tomcat/bin/nuxeo.conf
Data directory:
/Applications/nuxeo-dm-5.4.0.2-tomcat/nxserver/data
```

Copyright © 2001-2011 Nuxeo SAS (http://nuxeo.com/) and contributors.

The Control Panel gives you access to:

- A summary of the server status: is it running, is stopped, etc...
- The logs of the server: the console and server logs are information of the tasks the server is doing and messages on how it is processing these tasks.
- The Nuxeo Shell: the administrators' Swiss Army knife.

Here are the different ways to start and stop your Nuxeo application, depending on your OS:

Starting Your Nuxeo Application

By default, you cannot run two Nuxeo applications at the same time. If you want to run two Nuxeo applications at the same time (for instance a Nuxeo DM and a Nuxeo DAM), you need to change the default port used by one of the Nuxeo servers.

Depending on your OS, there are different ways to start the application. The steps below show how to use the Control Panel to start the server. However, you can use the command below in a terminal if you prefer. From the `$NUXEO_HOME/bin`, execute `nuxeoctl start`. You can refer to the nuxeoctl and Control Panel Usage for more information on the `nuxeoctl` command and the Control Panel.

Starting Your Application on Windows

1. Open the Nuxeo Control Panel:
    - In the folder `C:\Nuxeo application`, double-click on `Start Nuxeo.bat`.
    - In the folder `C:\Nuxeo application\bin`, double-click on `nuxeoctl.bat`.
      The Nuxeo Control Panel opens.
2. Click on the **Start** button.
   Starting the Nuxeo server takes between a few seconds and several minutes, depending on your hardware and the distribution you have chosen to install.
   When the server is started, the **Start** button becomes a **Stop** button.
3. Open a browser and type the URL http://localhost:8080/nuxeo/.

If the server is started for the first time after the installation, the startup wizard is displayed so you can select what module you want to install on the platform and help you configure it.
Otherwise, the login page is displayed so you can use the application.

> On Windows 7, you need to run the `nuxeoctl.bat` and `Start Nuxeo.bat` commands as an administrator if you haven't installed your Nuxeo application at the root of `C:` (for instance in `C:\Program Files`). To run them as an administrator, right-click on the command and click on "Run as administrator".

On Windows, it is possible to start Nuxeo as a service. Please report the Installing the Nuxeo Platform as a Windows Service page for guidelines and examples.

Starting Your Application on Linux

Nuxeo applications are started using scripts.

1. Launch a terminal and go to your installation directory.
2. Start the server using the `nuxeoctl` script (located in the `bin` directory):

   ```
   ./bin/nuxeoctl gui
   ```

   > The command used to launch the Control Panel may not be executable by default. If it is the case, in the terminal go to the `bin` directory of Nuxeo and type the line below to be able to use it:
   > `chmod +x *.sh *ctl`

   The Control Panel opens.
3. Click on the **Start** button.
   Starting the Nuxeo server takes between 30 sec and several minutes, depending on your hardware and the distribution you have chosen to install.
   When the server is started, the **Start** button becomes a **Stop** button.
4. Open a browser and type the URL http://localhost:8080/nuxeo/.
   If the server is started for the first time after the installation, the startup wizard is displayed so you can select what module you want to install on the platform and help you configure it.
   Otherwise, the login page is displayed so you can use the application.

Starting Your Application on Mac OS X

Mac OS users can use either the same steps as Linux users or some Mac OS convenient commands (see below).

1. From the Finder, click on "Start Nuxeo.command". You can also drag and drop the start script in the terminal and press Enter.

   > The command may not be executable by default. If it is the case, in the terminal go to the `bin` directory of Nuxeo and type the line below:
   > `chmod +x *.command`

   The Control Panel opens.
2. Click on the **Start** button.
   Starting the Nuxeo server takes between 30 sec and several minutes, depending on your hardware and the distribution you have chosen to install.
   When the server is started, the **Start** button becomes a **Stop** button.
3. Open a browser and type the URL http://localhost:8080/nuxeo/.
   If the server is started for the first time after the installation, the startup wizard is displayed so you can select what module you want to install on the platform and help you configure it.
   Otherwise, the login page is displayed so you can use the application.

Stopping Your Nuxeo Application

The steps to stop your Nuxeo application are the same for all operating systems.

To stop your server:

1. On the Control Panel, click on the **Stop** button.
   Stopping the server takes several seconds. When the server is stopped, the **Stop** button becomes a **Start** button.
2. Close the Control Panel.

If you started the server using the `nuxeoctl start` command in the terminal, use the `nuxeoctl stop` command to stop it.

nuxeoctl and Control Panel Usage

`nuxeoctl` Usage

The `nuxeoctl` script enables various options and commands (explained in details below).

Here is the Shell/Batch script usage:

```
nuxeoctl [options] <command> [command parameters]
```

Issue "`nuxeoctl help`" to print this information.
The options and command parameters between square brackets are optional.
The values separated by "|" are choices ("|" means "exclusive or").
You can use multiple options at once but only one command.

See the Environment variables page for setting Nuxeo Home and Configuration paths.

In this section

Options

Option	Description
`--accept=true|false|ask`	(Since Nuxeo 5.6) Accept, refuse or ask confirmation for all changes (default: `ask`) `--accept=true` also sets `--relax=true` (i.e. non interactive mode)
`-d` `--debug`	(Since Nuxeo 5.5) Activate debug messages. See '-dc' option.
`-dc <arg>`	(Since Nuxeo 5.6) Comma separated root categories for 'debug' option (default: "org.nuxeo.launcher").
`--gui=true|false`	(Since Nuxeo 5.6) Use graphical user interface (default is `true` on Windows and `false` on other platforms)
`-h` `--help`	Show detailed help
`--json`	(Since Nuxeo 5.6) Output JSON for mp-commands

`--nodeps`	(Since Nuxeo 5.6) Ignore package dependencies and constraints (old behavior)
`-q` `--quiet`	(Since Nuxeo 5.5) Activate quiet mode.
`--relax=true|false|ask`	(Since Nuxeo 5.6) Allow relax constraint on current platform (default: `ask`)
`--xml`	(Since Nuxeo 5.6) Output XML for mp-commands

Commands

Command	Description
`help`	Print this message.
`gui`	(Deprecated since Nuxeo 5.6: use `--gui` option instead) Use graphical user interface. On Linux/Mac OS X, default is in headless/console mode. On Windows, the `--gui=true` option is activated by default.
`nogui`	(Deprecated since Nuxeo 5.6: use `--gui` option instead) Windows only. This option deactivates the `gui` option which is set by default under Windows.
`start`	Start Nuxeo server in background, waiting for effective start. Useful for batch executions requiring the server being immediately available after the script returned.
`stop`	Stop any Nuxeo server started with the same `nuxeo.conf` file.
`restart`	Restart Nuxeo server.
`configure`	Configure Nuxeo server with parameters from `nuxeo.conf`.
`wizard`	Enable the wizard (force the wizard to be played again in case the wizard configuration has already been done).
`console`	Start Nuxeo server in a console mode. `Ctrl+C` will stop it.
`status`	Print server status (running or not).
`startbg`	Start Nuxeo server in background, without waiting for effective start. Useful for starting Nuxeo as a service.
`restartbg`	Restart Nuxeo server with a call to `startbg` after `stop`.
`pack`	Build a static archive (the "pack" Shell script is deprecated).
`showconf`	Display the instance configuration.
`mp-list`	List local Marketplace packages.
`mp-listall`	List all Marketplace packages (requires a registered instance).
`mp-init`	Pre-cache Marketplace packages locally available in the distribution.
`mp-update`	Update cache of Marketplace packages list.
`mp-add`	Add Marketplace package(s) to local cache. You must provide the package file(s), name(s) or ID(s) as parameter.
`mp-install`	Run Marketplace package installation. It is automatically called at startup if `installAfterRestart.log` file exists in data directory. Else you must provide the package file(s), name(s) or ID(s) as parameter.

`mp-uninstall`	Uninstall Marketplace package(s). You must provide the package name(s) or ID(s) as parameter (see "mp-list" command). If uninstalling a package by its ID and other versions of the same package are available, the most up-to-date will be installed instead.
`mp-remove`	Remove Marketplace package(s). You must provide the package name(s) or ID(s) as parameter (see "mp-list" command).
`mp-reset`	Reset all packages to DOWNLOADED state. May be useful after a manual server upgrade.
`mp-purge`	Uninstall and remove all packages from the local cache.
`mp-hotfix`	Install all the available hotfixes for the current platform (requires a registered instance).
`mp-upgrade`	Get all the available upgrades for the Marketplace packages currently installed (requires a registered instance).
`mp-show`	(Since Nuxeo 5.7.1) Show Marketplace package(s) information. You must provide the package file(s), name(s) or ID(s) as parameter.

Most `mp-*` commands will have different behavior if the instance is registered or not (they need an authenticated access to the private Marketplace channels).
If the server has no access to Internet, `mp-*` commands will only use packages available in the local cache.
If using a Marketplace package not compliant with the current platform, you will have to relax the constraint on current platform (see `--relax` option). Be very careful since that can lead to install other packages not designed for your current Nuxeo version unless you perform a unitary install (see `--nodeps` option).

Additional Parameters

All parameters following a command which accepts no parameter are passed to the Java process when executing the command.
That can be used for specific installs on which you rely on server specific parameters.

Java Usage

Launcher can be run as a Java command, without using the Shell (`nuxeoctl`) or Batch (`nuxeoctl.bat`) script.
The equivalent Java command to Shell command is printed at startup. See the line starting with "Launcher command:". It can be reused for writing your own scripts.

Here is the Java usage:

```
java [-Dlauncher.java.opts="JVM options"] [-Dnuxeo.home="/path/to/nuxeo"]
[-Dnuxeo.conf="/path/to/nuxeo.conf"] \
     [-Djvmcheck=nofail] -jar "path/to/nuxeo-launcher.jar" \
     [options] <command> [command parameters]
```

Java Options

Option	Description
`launcher.java.opts`	Parameters for the server JVM (default are -Xms512m -Xmx1024m -XX:MaxPermSize=512m).
`nuxeo.home`	Nuxeo server root path (default is parent of called script).
`nuxeo.conf`	Path to nuxeo.conf file (default is `$NUXEO_HOME/bin/nuxeo.conf`).
`jvmcheck`	If equals to "nofail", will continue execution even if JVM does not fit requirements, else will exit.

`gui`	Launcher with a graphical user interface. On Linux/Mac OS X, default is in headless/console mode. On Windows, the `gui` option is activated by default.

Commands

See the `nuxeoctl` commands.

Additional parameters

See the `nuxeoctl` additional parameters.

Exit Code Values

Exit code values are following the Linux Standard Base Core Specification 4.1.

If the status command was requested, nuxeoctl will return the following exit status codes:

0	program is running or service is OK
3	program is not running
4	program or service status is unknown

In case of an error while processing any action except for status, nuxeoctl shall print an error message and exit with a non-zero status code:

1	generic or unspecified error
2	invalid or excess argument(s)
3	unimplemented feature
4	user had insufficient privilege
5	program is not installed
6	program is not configured
7	program is not running

Troubleshooting

- Launcher Says It Couldn't Retrieve Process ID — [org.nuxeo.launcher.NuxeoLauncher] Sent server start command but could not get process ID.

Launcher Says It Couldn't Retrieve Process ID

Error Message

```
[org.nuxeo.launcher.NuxeoLauncher] Sent server start command but could not get process
ID.
```

If you got such a message in the console or in the console.log file, that means the Launcher was not able to confirm the server status, retrieving the process id.

Until Nuxeo 5.6, that warning message can also be due to a JVM start failed (for instance, not enough memory). You can check this by running `nuxeoctl console` instead of `nuxeoctl start`.

Since Nuxeo 5.7 (NXP-11039), the errors detection and associated messages were improved so that warning message only appears when the Java process is effectively started.

That can be a normal behavior on some old Windows versions and on Unix Solaris.

Windows users, if you can issue the two following commands without error in a Shell windows, then the Launcher must be able to manage the server process on your OS:

```
wmic quit
taskkill /?
```

Workarounds

Console Mode

First, you can start in console mode (in which case, the Launcher won't try to manage the process) in order to manually check if the server can start: `nuxeoctl console`. You will have to stop the server issuing `"CTRL+C"`.

If you can start the server that way, it should also be able to start with `nuxeoctl startbg`. The drawback is `nuxeoctl stop` won't be able to stop the server.

Java Commands

In case the issue lies in the batch file, you can try to run the Launcher with its Java command. If you issued `nuxeoctl start` from a Shell window, then you can copy/paste the command titled "Launcher command". It will be of the form:

Launcher Java command

```
java -Dlauncher.java.opts="some JVM options" -Dnuxeo.home=/path/to/nuxeo/
-Dnuxeo.conf=/path/to/nuxeo.conf -jar /path/to/nuxeo-launcher.jar start
```

You can also try to directly run the server, fully bypassing the Launcher. Look into the `console.log` file for a command titled "Server command". You will have to run the Launcher configuration, then the server command. It will look like:

Run the server, bypassing the Launcher

```
nuxeoctl configure
java -cp "the classpath" -Dnuxeo.home=/path/to/nuxeo -Dnuxeo.conf=/path/to/nuxeo.conf
-Dnuxeo.log.dir=log -Dnuxeo.data.dir=data -Dnuxeo.tmp.dir=tmp
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
-Dcatalina.base=/path/to/nuxeo -Dcatalina.home=/path/to/nuxeo
org.apache.catalina.startup.Bootstrap start
```

Debugging

Then, if you want to understand the failure cause, you can try to manually get that process ID from a Shell window, reproducing the steps used by the Launcher.

Look into `console.log` for the "regexp" keyword, this is the regular expression used to find the process ID. Something like:

```
[org.nuxeo.launcher.NuxeoLauncher] regexp:
\Q/path/to/nuxeo.conf\E.*\Qorg.apache.catalina.startup.Bootstrap\E pid:null
```

The command for retrieving the process ID will depend on your OS.

Linux/Unix

```
/bin/ps -e -o "pid,args" | grep -E "^\s*([0-9]+)\s+(.*)$" | grep
"/path/to/nuxeo.conf.*org.apache.catalina.startup.Bootstrap"
```

Mac OS X

```
/bin/ps -e -o "pid,command" | grep -E "^ *([0-9]+) +(.*)$" | grep
"/path/to/nuxeo.conf.*org.apache.catalina.startup.Bootstrap"
```

Windows

```
wmic process get CommmandLine,ProcessId
```

Look for lines matching the regular expression: `^(.*?)\\s+(\\d+)\\s*\$`

Check if one of those is matching the regular expression found in `console.log`.

Monitoring and Maintenance

Monitoring Nuxeo

Nuxeo JMX Monitoring

Nuxeo platform exposes counters, probes and stopwatch via nuxeo-runtime-management.

Nuxeo Server Running and Components Loading Statuses

Since 5.5, Nuxeo provides an URL for monitoring the server status. This method is actually also used by the Launcher to follow the server startup status, after checking the Java process status.
http://localhost:8080/nuxeo/runningstatus will be available at last. While it isn't reachable, the server is stopped or still starting.
http://localhost:8080/nuxeo/runningstatus?info=started returns `true` if the server finished starting and the Nuxeo runtime is fine with its components.
http://localhost:8080/nuxeo/runningstatus?info=summary&key=xxx returns `true` or `false` (see "info=started") and a detailed summary about components. Access to this URL is restricted by an access key configurable in `nuxeo.conf` (see `"server.status.key"` in Configuration Parameters Index (nuxeo.conf)).

Sample output if something was wrong at startup (for instance, missing RelationService)

```
false
======================================================================
= Nuxeo EP Started
======================================================================
= Component Loading Status: Pending: 7 / Unstarted: 0 / Total: 462
  * service:org.nuxeo.ecm.webengine.sites.wiki.relation requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.annotations.graph requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.platform.relations.jena requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.annotations.repository.graph requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.platform.publisher.relations.contrib requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.platform.relations.services.DefaultJenaGraph requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
  * service:org.nuxeo.ecm.platform.comment.service.CommentServiceConfig requires
[service:org.nuxeo.ecm.platform.relations.services.RelationService]
======================================================================
```

In this section

You can get that information with `./bin/nuxeoctl status` (see nuxeoctl and Control Panel Usage).

JVM Garbage Collector

The garbage collector attempts to reclaim memory used by objects that are no longer in use by the application.

Monitoring the garbage collector can be very useful when tuning the JVM or setting the initial heap size.

Edit `$NUXEO_HOME/bin/nuxeo.conf` and uncomment the following options titled "Log Garbage Collector informations into a file":

```
JAVA_OPTS=$JAVA_OPTS -Xloggc:${nuxeo.log.dir}/gc.log -verbose:gc -XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
```

JBoss

The JBoss LoggingMonitor service can monitor specific attributes of a MBean periodically and log their value to the filename specified.

More info on the LoggingMonitor:http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossLoggingMonitor

Edit `$NUXEO_HOME/bin/nuxeo.conf` and add `"monitor"` to the `nuxeo.templates` parameter (uncomment it if needed).

The logging-monitor jar file is deployed by default in $JBOSS_HOME/server/default/lib.

JBoss JVM Information

The `"monitor"` template will deploy a file named `jvm-monitor-service.xml` which will produce a `jvm.log` file.

JBoss Thread Pool

The `"monitor"` template will deploy a file named `webthreads-monitor-service.xml` which will produce a `webthreads.log` file.

Nuxeo Unified Datasource Connection Pool

The `"monitor"` template will deploy a file named `default-ds-monitor-service.xml` which will produce a `nuxeo-ds.log` file.

PostgreSQL

The PostgreSQL logs can be setup like in the pgFouine tutorial:

http://pgfouine.projects.postgresql.org/tutorial.html

For instance to log only request slower than 100ms change your postgresql.conf file:

```
log_min_duration_statement = 100
```

After a test you can catch the vacuum output like this:

```
vacuumdb  -fzv $DBNAME &> vacuum.log
```

OS

The sysstat utilities are a collection of performance monitoring tools for Linux that is easy to setup.

You can monitor the system activity like this:

```
sar -d -o $JBOSS_HOME/server/default/log/sysstat-sar.log 5 720 >/dev/null 2>&1 &
```

This will monitor the activity every 5s during 1h.

For more information on systat, visit `http://pagesperso-orange.fr/sebastien.godard/`.

Log Analysis

logchart

This is a small script that process the following log files:

- Garbage collector logging (`gc.log`)
- Java Virtual Machine logging (`jvm.log`)
- JBoss threads logging (`webthreads.log`)
- NuxeoDS Data source usage (`nuxeo-ds.log`)
- Sysstat sar logging, cpu and disk activity (`sysstat-sar.log`)
- PostgreSQL logs (`pgsql.log`)
- PostgreSQL vacuum output (`vacuum.log`)

to generate an html report with charts:
`http://public.dev.nuxeo.com/~ben/logchart/monitor.html`

More information on logchart can be found on the `README.txt` of the project:

`https://hg.nuxeo.org/tools/logchart/trunk`

Other Reporting Tools

- GCViewer a tool to visualize data produced by the garbage collector logs: http://www.tagtraum.com/gcviewer.html
- kSar a sar grapher that can produce detail PDF report of your system activity: http://ksar.atomique.net/linux.html
- pgfouine the PostgreSQL log analyzer which is used by logchart: http://pgfouine.projects.postgresql.org/

Nuxeo Metrics

Since Nuxeo 5.7.1, the platform uses Coda Hale Yammer Metrics.

These metrics are exposed via JMX but can also be reported with CSV files or send to a Graphite server.

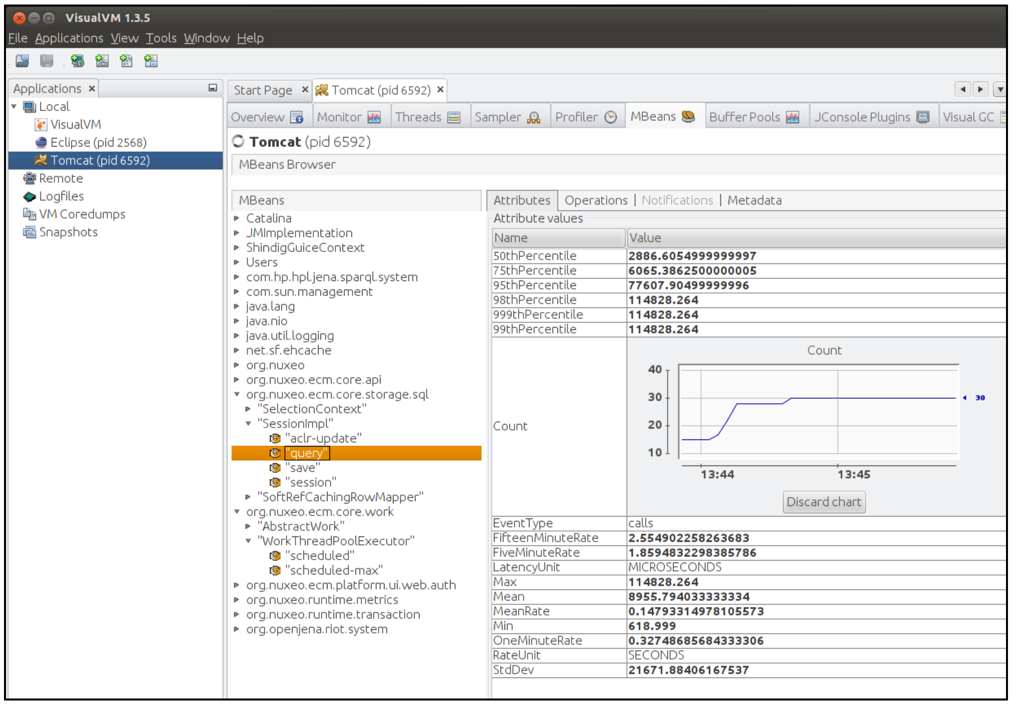
Reporting Metrics

Enable JMX Reporting

To enable JMX reporting add the following to the nuxeo.conf file (warning this is not secure):

```
JAVA_OPTS=$JAVA_OPTS -Dcom.sun.management.jmxremote=true
```

In this section



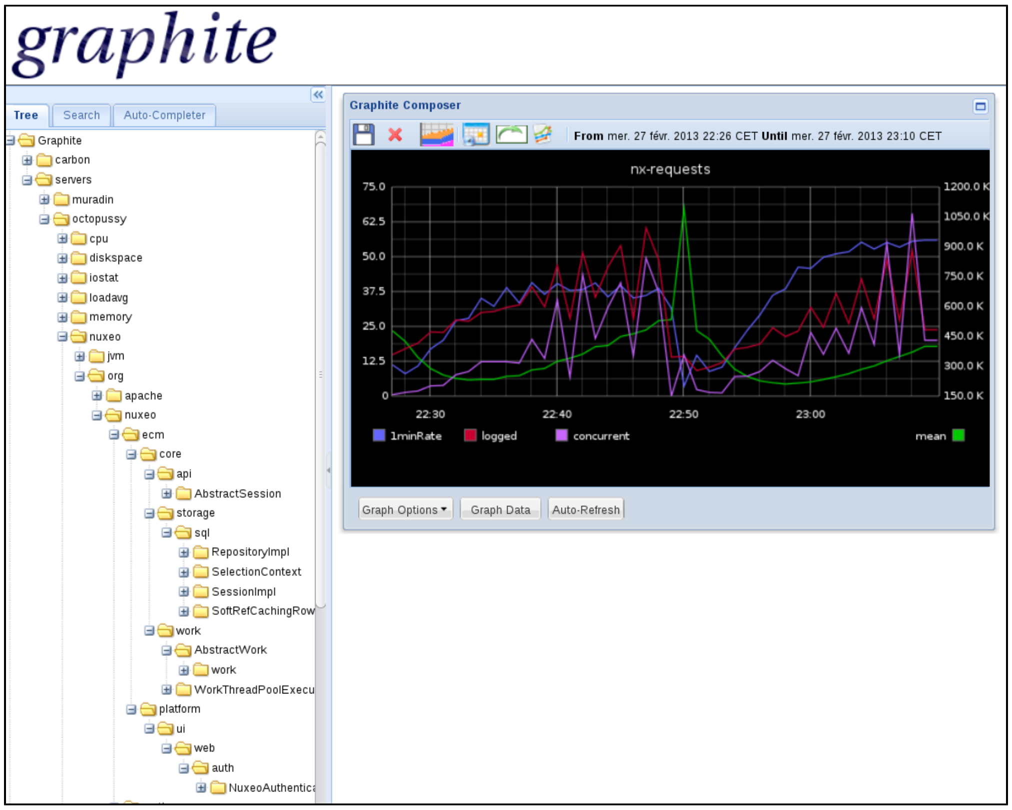
Name	Value
50thPercentile	2886.6054999999997
75thPercentile	6065.3862500000005
95thPercentile	77607.90499999996
98thPercentile	114828.264
999thPercentile	114828.264
99thPercentile	114828.264
Count	
EventType	calls
FifteenMinuteRate	2.554902258263683
FiveMinuteRate	1.8594832298385786
LatencyUnit	MICROSECONDS
Max	114828.264
Mean	8955.794033333334
MeanRate	0.14793314978105573
Min	618.999
OneMinuteRate	0.3274868568433306
RateUnit	SECONDS
StdDev	21671.88406167537

Enabling CSV reporting

```
metrics.csv.enabled=true
metrics.csv.period=10
# This will create a sub directory metrics-TIMESTAMP
metrics.csv.dir=${nuxeo.log.dir}
```

Enabling Graphite reporting

```
metrics.graphite.enabled=true
metrics.graphite.host=localhost
metrics.graphite.port=2030
metrics.graphite.period=10
```



Reporting log4j stats

```
metrics.log4j.enabled=true
```

Reporting tomcat JMX info:

```
metrics.tomcat.enabled=true
```

Note that period to report metrics are in second.

Nuxeo Metrics

Metrics are prefixed by default with **`servers.${HOSTNAME}.nuxeo`** to be compliant with Diamond prefix, this can be changed by setting the **`metrics.graphite.prefix`** in nuxeo.conf.

- `prefix.nuxeo.org.nuxeo.ecm.core.api.AbstractSession`
    - create-document: Counter of document created
    - delete-document: Counter of document deleted
    - update-document: Counter of document updated
- `prefix.nuxeo.org.nuxeo.ecm.core.storage.sql.RepositoryImpl`
    - session: Counter of VCS Session
- `prefix.nuxeo.org.nuxeo.ecm.core.storage.sql.SelectionContext`
    - cache-get: Timer on selection cache get operation
    - cache-hit: Counter on cache hit

    - cache-size: Size of the cache
- `prefix.nuxeo.org.nuxeo.ecm.core.storage.sql.SelectionImpl`
    - aclr-update: Timer on read acl optimization update
    - query: Timer on query operation (session.query and session.queryAndFetch)
    - save: Timer on session save operation
    - session: Counter of VCS session
- `prefix.nuxeo.org.nuxeo.ecm.core.storage.sql.SoftRefCachingRowMapper`
    - cache-get: Timer on row cache get
    - cache-hit: Counter on cache hit
    - cache-size: Counter on cache size (may be inaccurate)
    - sor-get: Timer on non cache get (System Of Record = db)
    - sor-rows: Counter on number of rows returned by the sor access
- `prefix.nuxeo.org.nuxeo.ecm.core.storage.work.AbstractWork`
    - work: Timer on work executions
- prefix.nuxeo.org.nuxeo.ecm.core.storage.work.WorkThreadPoolExecutor
    - scheduled: Counter on work pending
    - scheduled-max: Counter that keep the max of pending worker
- `prefix.nuxeo.org.nuxe.ecm.platform.ui.web.auth.NuxeoAuthentificationFilter`
    - logged-user: Counter of logged in user
    - request: Timer on request
    - request-concurrent: Counter of concurrent requests
    - request-concurrent-max: Max of concurrent requests
- `prefix.nuxeo.org.nuxeo.ecm.runtime.metrics.MetrcisServiceImpl`
    - instance-up: 1 if instance is up, 0 on shutdown
    - jdbc-numActive: jdbc/nuxeo datasource pool numActive connection
    - jdbc-numIdle: jdbc/nuxeo datasource pool numIdle connection
    - tomcat-activeSessions: tomcat activeSessions
    - tomcat-currentThreadCount: tomcat http connector thread pool currentThreadCount
    - tomcat-currentThreadBusy: tomcat http connector thread pool currentThreadBusy
    - tomcat-errorCount: tomcat errorCount
    - tomcat-processingTime: tomcat processingTime
    - tomcat-requestCount: tomcat requestCount
- `prefix.nuxeo.org.nuxeo.ecm.runtime.transaction.TransactionHelper`
    - rollback: Counter of rollback transaction
    - transaction: Timer on transaction
    - tx-concurrent: Counter of concurrent transaction
    - tx-concurrent-max: Max concurrent transaction

If you have enable log4j metrics you will have counters on log per severity (ERROR, WARN, ...)

- `prefix.nuxeo.org.apache.log4j.Appender.*`

The Graphite reporter also include JVM metrics for GC, memory and thread pool.

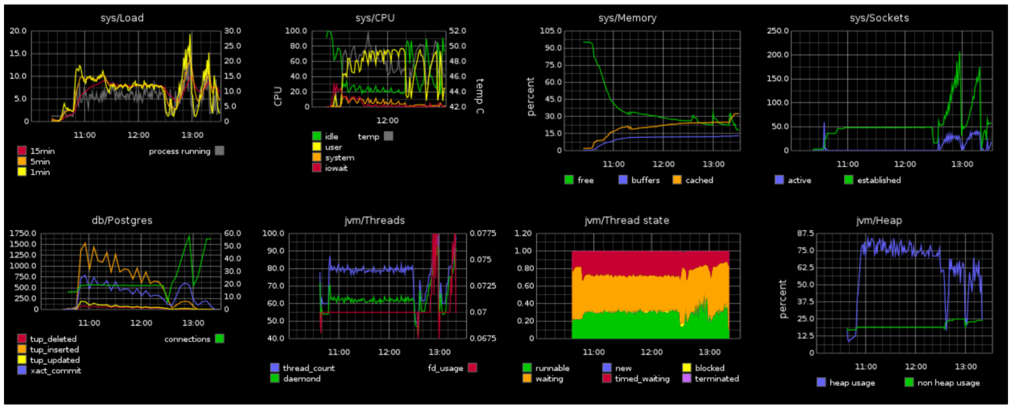
You should also monitor the system and the database to have a complete monitoring, tools like Diamond can do this easily.

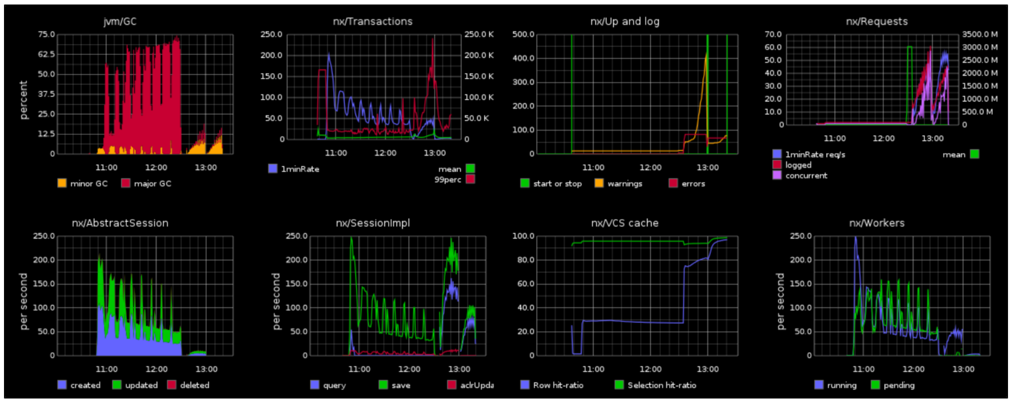
Graphite Dashboard

You can find an example of Graphite dashboard on GitHub: https://github.com/nuxeo/nuxeo-runtime/blob/release-5.8/nuxeo-runtime-metrics/graphite/dashboard.json.

You will have to edit the dashboard to replace the hostname (here it is octopussy).

Here is an extract of what this dashboard looks like when monitoring a daily bench.





Related topics

- Nuxeo Metrics (Nuxeo Installation and Administration - 5.8)
- Logs Analysis (Nuxeo Installation and Administration - 5.8)
- Setup (Nuxeo Installation and Administration - 5.8)
- Configuration Parameters Index (nuxeo.conf) (Nuxeo Installation and Administration - 5.8)

Backup and Restore

Backing Up

Nuxeo supports hot backup of your data.

If you have followed the recommendations, then you have configured Nuxeo to use a production-safe database (instead of the default Derby) and have set a path for `nuxeo.data.dir` in your `nuxeo.conf`. In that case:

1. Simply first backup your database (make a SQL dump),
2. Then backup your data on filesystem.

Performing the backup in that order (the database first, then the filesystem) will ensure backup consistency.

If you didn't configure Nuxeo to use a database, then the default database is embedded in the data directory: Stop the server before backup.

If you didn't configure Nuxeo data directory (`nuxeo.data.dir` in `nuxeo.conf`), then you have to find the default path which depends on the server (Tomcat/JBoss) and the Nuxeo version (look at the data directory value in the Admin Center).

- For Tomcat, it should be `$TOMCAT/nxserver/data`.
- For JBoss, it should be `$JBOSS/server/default/data`.

Restoring

1. Restore the database and data filesystem you had previously backed up.
2. Configure Nuxeo to use those database and data directory.
3. Start Nuxeo.

Additional Information

Two elements allow saving the filesystem once the database has been dumped:

- When you add a document in the repository, VCS computes the digest of the blob: it is this digest which is used as the filename of the document stored in the filesystem. That way, if a user uploads a different document but which has the same filename, the blob stored on filesystem won't be changed: a new blob with a different digest will be put in the blobstore.
- Blobs are not deleted as soon as the document is removed from the repository.

These two points ensures that no data will be modified (or deleted) after dumping your database. Only creation could happen. So the backup of the filesystem will be consistent with the backup of the database.

Some remarks about VCS:

- As VCS uses the digest of the blob, this ensures a document will be stored only once in the blobstore, even if it is uploaded several times.
- As VCS doesn't delete blobs once a document is removed from the repository, you should run a clean-up regularly from the Admin Center in the menu **System Information / Repository binaries**.

Upgrading your Nuxeo Version

Each version of Nuxeo comes with bug fixes, new features and improvements.

This means upgrading to the latest public release is a smart move. In order to make upgrade easy, we are very careful not to break anything.

Remember that we provide support and use the Nuxeo Platform in a lot of projects, so if we break something, we have to fix it :)

So basically:

- We don't break the APIs between two versions: we add new APIs and deprecate old ones.
- There are several minor versions between the deprecation and the removal so you have time to adapt your code.
- If we completely replace a service (that was the case for `SearchService` and `EventService` for example), we provide compatibility packages so you can continue using the old API (even if migrating to the new API is highly recommended).

In terms of Data Migration we are also very careful not to break anything. Once again, we run the Nuxeo Platform for a lot of our internal needs and we upgrade them very frequently and don't want to have data migration issue.

Anyway, when some changes or optimizations impact the storage layer we either:

- make the upgrade automatically,
- or provide guidelines for the upgrade.

Upgrading is usually a simple and painless process.

Using the template system also allows to easily transpose your configuration from one version to an other.

In the worse cases, in case of problem, Nuxeo Support is there to help you

For upgrade steps, check the following documentation where you will find upgrade procedures per Nuxeo version:

- How to Upgrade Nuxeo
- How to Replicate the Nuxeo Repository
- Nuxeo Security Hotfixes

How to Upgrade Nuxeo

This document describes a general upgrade procedure, check if your version needs particular action.

See Upgrading your Nuxeo Version for information about Nuxeo policy on development and versions.

General Upgrade Procedure

You should have configured Nuxeo with a specific configuration (and optionally with custom templates), setting a database (for production) and a data directory. In that case, upgrading Nuxeo simply consists in:

1. Doing a backup.
2. Stopping the old Nuxeo.
3. Deploying the new distribution.
4. Updating the environment variables (`NUXEO_HOME` and `NUXEO_CONF`) to make it use your custom configuration, database and data directory.
   To update the `NUXEO_CONF` you should report your custom configuration (uncommented lines in `nuxeo.conf`) into the nuxeo.conf file of the target Nuxeo version, and then replace the old nuxeo.conf file with this new one.
5. Running `nuxeoctl mp-reset` in order to reset all Marketplace packages to the DOWNLOADED state.
6. Starting the new Nuxeo.

Upgrade Path

You will find below specific cases and required manual steps per version. Follow them carefully from your current version to the new version.
For instance, if upgrading from 5.3.2 to 5.4.2, follow steps about 5.3.2->5.4.0, 5.4.0->5.4.1 and 5.4.1->5.4.2.
If there are no specific instructions, then the upgrade will be smooth. For instance, if upgrading from 5.4.1 to 5.4.2, you have nothing to do unless you are using Oracle.

Upgrade notes
Although major changes are explicitly documented, it is recommended to look at the upgrade notes before upgrading. Upgrade notes list changes that may affect existing configuration or features after upgrade (changes on parameters, API, default behavior, ...).

In this section

Data

Following the upgrade path should guarantee a valid database structure once all upgrades are done. Should you have to perform some specific actions, they will be indicated (see Upgrade from 5.4.1 to 5.4.2 with Oracle for instance).

If you didn't configure Nuxeo to use a database, then the default database is embedded in the `data` directory.

If you didn't configure Nuxeo data directory (`nuxeo.data.dir` in `nuxeo.conf`), then you have to find the default path which depends on the server (Tomcat/JBoss) and the Nuxeo version (look at the `data` directory value in the Admin Center):

- With Tomcat, it should be `$TOMCAT/nxserver/data`.
- With JBoss, it should be `$JBOSS/server/default/data`.

Custom Code

As explained on the Upgrading your Nuxeo Version page, we are careful not to break the compatibility between versions. However, since the Platform evolves, you will also need to upgrade your custom code.

So the first step to do so is to identify:

- The contributions to the extension points you defined: this should be the easiest part because we try to keep compatibility between the versions;
- The XHTML templates from the Nuxeo Platform you overrode to get a custom behavior: some of these templates may have changed (sometimes a lot if you plan to upgrade several versions up) and you have to identify the modifications you did before in order to apply them to the new version of these Nuxeo templates;
- The API you used in your custom Java code: some APIs were deprecated, other ones removed (like everything around jBPM).

Then, you have to build your custom plugin against each version of the upgrade path and check if it's running correctly once deployed in a Nuxeo Platform.

To 5.8 from 5.6 or 5.7.x

5.8 release notes.

5.7.3 upgrade notes.

5.7.2 upgrade notes.

5.7.1 upgrade notes.

See Upgrade from 5.6 to 5.8.

To 5.6 from 5.5

Upgrade notes.

See Upgrade from 5.5 to 5.6.

To 5.5 from 5.4.2

Upgrade notes.

See Upgrade from 5.4.2 to 5.5.

To 5.4.2 from 5.4.1

Upgrade notes.

Oracle

If using Oracle, see Upgrade to 5.4.2 with Oracle.

To 5.4.1 from 5.4.0.1

Upgrade notes.

To 5.4.0.1 from 5.4.0

Upgrade notes.

To 5.4.0 from 5.3.2

Upgrade notes.

JBoss

If using the JBoss distribution, see Upgrade to 5.4 and JBoss 5.

Workflow Feature

The workflow implementation has changed, see From the old workflow system to the new 5.4 workflow system.

To 5.3.2 from 5.3.1

Upgrade notes.

See Upgrade from 5.3.1 to 5.3.2.

MySQL

If using MySQL, see Upgrade from 5.3.1 with MySQL to 5.3.2.

To 5.3.1 from 5.3.0

Upgrade notes.

See Upgrade from 5.3.0 to 5.3.1.

To 5.3.0 from 5.2.0

Upgrade notes.

To 5.2.0 from 5.1.6

Upgrade notes.

If using JCR with PostgreSQL, see How to migrate a Nuxeo 5.1.6 to Nuxeo 5.2 under JCR+PostgreSQL configuration.

To 5.1.6 from 5.1.3

Upgrade notes.

To 5.1.3 from 5.1.2

No upgrade notes.

See Upgrade from 5.1.2 to 5.1.3.

To 5.1.x from Earlier

1. First, stop Nuxeo EP and, before any upgrade, always backup (at least)
   - `$JBOSS/server/default/lib/nuxeo*`
   - `$JBOSS/server/default/deploy/nuxeo.ear/`
   - `$JBOSS/server/default/data/`
   - dump your database(s) if you have any.
2. If you have specific configuration, back up separately the configuration files in order to easily re-apply them onto the default one (take care not to lose any modification on these files):
   - `$JBOSS/server/default/deploy/nuxeo.ear/config/`
   - `$JBOSS/server/default/deploy/nuxeo.ear/datasources/`
   - `$JBOSS/server/default/deploy/nuxeo.ear/platform/nuxeo-platform-search-compass-plugin*/compass.cfg.xml`
   - any other JBoss files you could have changed (`mail-service.xml`, etc.).
3. It could also be useful to back up logs from $JBOSS/server/default/log/.
4. Move your JBoss directory and replace it with the one coming from the new Nuxeo EP version.
5. Replace `$JBOSS/server/default/data/` with your backup data.
6. Apply again you configuration.
7. Start Nuxeo.

Upgrade from 5.6 to 5.8

For the general upgrade process, see the How to Upgrade Nuxeo page.
This chapter highlights some major information about upgrade from Nuxeo Platform 5.6 to Nuxeo Platform 5.8. Most of it is useful information you need to have to fully understand what has changed in this release.
If you need an exhaustive list, see the 5.6 -> 5.8 upgrade notes.

You may also want to have a look at the 5.8 release notes.

If you have followed the Fast Track releases since Nuxeo Platform 5.6, you can refer to the Upgrade from 5.7.x to 5.8 section.

Installation & Configuration

Follow the Installation instruction.

Java 7 Required

Since version 5.7.1, the Nuxeo Platform enforces the use of Java 7 because it's faster (better GC among other things) and because Java 6 is getting unsupported, see NXP-11226.

Upgrade to Tomcat 7

Since version 5.7.2, the Nuxeo Platform was upgraded to Tomcat 7. Please follow the upgrade notes of NXP-10071.

On this page

Data migration

Relations

Since Nuxeo Platform 5.7.1 all the relations (including comments and publication relations, but excluding annotations) are stored by default in the VCS database. If you upgrade from a previous configuration (before 5.6) where you created relations, comments or publications, then these relation objects were stored using Jena. To keep using this configuration, please follow the upgrade notes of NXP-10350.

Derby Storage Moved to H2

For test and development instances that use the default configuration, audit logs storage has been moved from Derby to H2. If you plan on upgrading your test data on H2 from 5.6 to 5.8, note that audit logs will be lost unless you migrate data from Derby to H2 or change database configuration to use the old Derby database for audit logs.

This also applies to vocabularies, annotations, activities, local configuration and sequences.

SQL Server

The column type for ID columns has been changed from VARCHAR to NVARCHAR, for efficiency reasons. All new tables will be created with NVARCHAR, but existing tables must be migrated. Please follow the upgrade notes of NXP-10862.

Code Migration

EL expressions in Action Filters

From Nuxeo Platform 5.7.3, EL expressions in action filters are using a more generic context to allow better filtering expressions. Please follow the upgrade notes of NXP-10566.

Query Models and Result Provider Farms Removal

Content views and page providers have been in place since 5.4.2 to replace old query models and result provider farms, so old classes have been removed for 5.8.

Please follow detailed upgrade notes for these features.

Upgrade from 5.7.x to 5.8

Previous Fast Track upgrades notes:

- 5.7.1 upgrade notes
- 5.7.2 upgrade notes
- 5.7.3 upgrade notes
- 5.8 upgrade notes

Query Models and Result Providers Migration to Content Views and Page Providers

This page explains how query models and result provider farms, deprecated since version 5.4.2, can be migrated to content views and page providers.

The JIRA issue Unable to render JIRA issues macro, execution error. gives technical upgrade notes, this chapter gives more details depending on use cases.

Generic Migration Steps

Content views and associated page providers have been designed to replace old query models that were not as modular and easy to use.

A lot of configuration has been updated or removed, but without any feature loss. For instance, syndication links are not relying anymore on a specific module, but using native exports from the result layout listing.

Reading the Content Views chapter is highly recommended to understand migration steps.

Query models have been replaced by page providers. Page providers definition is sometimes embedded within a content view definition, as content views are useful for UI interactions. Page providers alone are still useful to generate lists of documents from core code, not involving any UI interactions. Results provider farms were used to pass UI contextual information to the query model. This is now done by using EL expressions when defining parameters on content views.

In this section

XML Configurations Migration

Components `org.nuxeo.ecm.core.search.api.client.querymodel.QueryModelService` and `org.nuxeo.ecm.webapp.pagination.ResultsProviderService` have been removed, components `org.nuxeo.ecm.platform.query.api.PageProviderService` (with extension point providers) and `org.nuxeo.ecm.platform.ui.web.ContentViewService` (with extension point contentViews) should be used instead.

The XML syntax is very close, here is a sample migration of a query model contribution without a whereClause element:

Old Configuration

```
<extension
target="org.nuxeo.ecm.core.search.api.client.querymodel.QueryModelService"
  point="model">
  <queryModel name="MY_SEARCH">
    <pattern>
      SELECT * FROM Document WHERE ecm:currentLifeCycleState != 'deleted' AND
ecm:uuid != ?
    </pattern>
    <sortable value="true" defaultSortColumn="dc:title" defaultSortAscending="true"
/>
    <max>20</max>
  </queryModel>
</extension>
```

This can be translated into a page provider very easily (notice the pageSize and sort syntax changes):

New Configuration with Page Provider

```
 <extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <coreQueryPageProvider name="MY_SEARCH">
    <pattern>
      SELECT * FROM Document WHERE ecm:currentLifeCycleState != 'deleted' AND
ecm:uuid != ?
    </pattern>
    <sort column="dc:title" ascending="true" />
    <pageSize>20</pageSize>
  </coreQueryPageProvider>
</extension>
```

Code Migration

Classes `QueryModel`, `QueryModelService` and `ResultsProviderFarm` have been removed.

The `QueryModel` class is basically replaced by `PageProvider` or `PageProviderDefinition` instances depending on the use case. Default page providers are available and perform queries on the Nuxeo repository, for instance CoreQueryDocumentPageProvider.

Reading the Custom Page Providers and Page Providers without Content Views chapters is recommended to understand how to use the new API.

Templates Migration

Old templates displaying paged lists of documents have been removed, the template at `/incl/content_view.xhtml` can now be included to display the results using a listing layout configured on the content view.

Sample Old Template

```
<div xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:nxu="http://nuxeo.org/nxweb/util">

  <nxu:set var="documents"
    value="#{documentActions.getChildrenSelectModel()}">
    <ui:decorate template="/incl/forum_table.xhtml" />
  </nxu:set>

</div>
```

Alternative Sample Old Template

```
<div xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:nxu="http://nuxeo.org/nxweb/util">

  <nxu:set var="provider"
    value="#{resultsProvidersCache.get('MY_SEARCH')}"
    cache="true">
    <ui:decorate template="/search/documents_table.xhtml">
      <ui:param name="provider" value="#{provider}"/>
      <ui:param name="providerName" value="#{provider.name}"/>
    </ui:decorate>
  </nxu:set>

</div>
```

Sample New Template

```
<div xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:nxu="http://nuxeo.org/nxweb/util">

  <nxu:set var="contentViewName" value="MY_SEARCH">
    <ui:decorate template="/incl/content_view.xhtml" />
  </nxu:set>

</div>
```

Also, the old templates displaying listings of documents were not relying on layouts, so migration may include defining listing layouts and widget templates when migrating to content views.

Migration Use Cases

Migrating a `QueryModel` to a `PageProvider`

Let's take again the above example:

Old Configuration

```
<extension
target="org.nuxeo.ecm.core.search.api.client.querymodel.QueryModelService"
  point="model">
  <queryModel name="MY_SEARCH">
    <pattern>
      SELECT * FROM Document WHERE ecm:currentLifeCycleState != 'deleted' AND
ecm:uuid != ?
    </pattern>
    <sortable value="true" defaultSortColumn="dc:title" defaultSortAscending="true"
/>
    <max>20</max>
  </queryModel>
</extension>
```

This query model is designed to perform a query on the Nuxeo Core Repository, using a parameter to fill the ecm:uuid filtering criterion.

Here is a sample JAVA code using this query model:

```
QueryModelService qmService = Framework.getLocalService(QueryModelService.class);
QueryModelDescriptor qmd = qmService.getQueryModelDescriptor("MY_SEARCH");
QueryModel qm = new QueryModel(qmd);
Object[] params = {document.getId()};
DocumentModelList list = qm.getDocuments(coreSession, params);
```

Let's migrate the query model to a page provider:

New Configuration with Page Provider

```
<extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <coreQueryPageProvider name="MY_SEARCH">
    <pattern>
      SELECT * FROM Document WHERE ecm:currentLifeCycleState != 'deleted' AND
ecm:uuid != ?
    </pattern>
    <sort column="dc:title" ascending="true" />
    <pageSize>20</pageSize>
  </coreQueryPageProvider>
</extension>
```

Let's also migrate the corresponding JAVA code:

```
PageProviderService ppService =
Framework.getLocalService(PageProviderService.class);
Map<String, Serializable> props = new HashMap<String, Serializable>();
props.put(CoreQueryDocumentPageProvider.CORE_SESSION_PROPERTY,
        (Serializable) coreSession);
Object[] params = {document.getId()};
PageProvider<DocumentModel> pp = (PageProvider<DocumentModel>)
ppService.getPageProvider(
        "MY_SEARCH", null, null, null, props, params);
DocumentModelList list = pp.getCurrentPage();
```

Here is a more complex migration involving a whereClause (the search pattern is generated according to predicates definitions, retrieving values on the search document model):

```
<extension
target="org.nuxeo.ecm.core.search.api.client.querymodel.QueryModelService"
  point="model">
  <queryModel name="MY_SEARCH" docType="AdvancedSearch">
    <whereClause>
      <predicate parameter="ecm:fulltext" operator="FULLTEXT ALL">
        <field schema="advanced_search" name="fulltext_all"/>
      </predicate>
      <fixedPart>ecm:currentLifeCycleState != 'deleted'</fixedPart>
    </whereClause>
    <sortable value="true" defaultSortColumn="dc:title" defaultSortAscending="true"
/>
    <max>20</max>
  </queryModel>
</extension>
```

The `whereClause` element content is unchanged, but the associated `docType` element has moved from the `queryModel` element to the `whereClause` element:

```
<extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <coreQueryPageProvider name="MY_SEARCH">
    <whereClause docType="AdvancedSearch">
      <predicate parameter="ecm:fulltext" operator="FULLTEXT ALL">
        <field schema="advanced_search" name="fulltext_all"/>
      </predicate>
      <fixedPart>ecm:currentLifeCycleState != 'deleted'</fixedPart>
    </whereClause>
    <sort column="dc:title" ascending="true" />
    <pageSize>20</pageSize>
  </coreQueryPageProvider>
</extension>
```

Migrating a `QueryModel` to a `ContentView`

Sometimes it is useful to migrate the query model to a content view, where the page provider definition is embedded.

Here is a sample migration of the above example to a content view:

```
<extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <contentView name="MY_SEARCH">
    <coreQueryPageProvider>
      <pattern>
        SELECT * FROM Document WHERE ecm:currentLifeCycleState != 'deleted' AND
ecm:uuid != ?
      </pattern>
      <sort column="dc:title" ascending="true" />
      <pageSize>20</pageSize>
      <parameter>#{currentDocument.id}</parameter>
    </coreQueryPageProvider>
    [...]
  </contentView>
</extension>
```

Notice the parameter element, using an EL expression, that makes it possible to resolve Seam/JSF EL expresssions.

Migrating a QueryModel associated to a ResultsProviderFarm to a ContentView

Result provider farms were useful to pass contextual parameters to the page provider. Using EL expressions as parameter elements in the definition as above makes it possible to map directly these parameters.

For instance, if the result provider farm is a seam component named `mySeamComponent` and is using one of its custom contextual field `myFi eld` as a parameter for the query model, the content view can simply state a parameter in the page provider definition as is (provided the Seam component holds public a getter method for this field):

```
<extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <contentView name="MY_SEARCH">
    <coreQueryPageProvider>
      [...]
      <parameter>#{mySeamComponent.myField}</parameter>
    </coreQueryPageProvider>
    [...]
  </contentView>
</extension>
```

More Migration Examples

Nuxeo source code has been upgraded too, here are sample changes that can be helpful as examples:

- Migration of the forum module, involving creation of a listing layout, see commit https://github.com/nuxeo/nuxeo-features/commit/acfccb24d306406e0ec66d85ec80f29b34f8cf02
- Migration of the sample module, involving migration of a query model and result provider farm, especially commit https://github.com/nuxeo/nuxeo-sample-project/commit/c8cb1a9dd7708ffc236780efc36db00032accf0e (see Unable to render JIRA issues macro, execution error.)
- Migration of the picture book listing view, involving a partial migration of XHTML templates to keep javascript custom code, see commits https://github.com/nuxeo/nuxeo-features/commit/85cafc1883275f337bf214fb3a79addc5e8cf5aa and https://github.com/nuxeo/nuxeo-features/commit/aa12896607d6acfe5dcd55cbc55652af789088bc.

Upgrade from 5.5 to 5.6

This chapter presents the detailed process to upgrade from Nuxeo 5.5 to Nuxeo 5.6. Most of it is useful information you need to have to fully understand what has changed in this release.

Installation & Configuration

Follow the Installation instruction.

The following lists known upgrade issues.

Database

Even if your custom template defines an inclusion of 'database' template,

```
nuxeo.template.includes=postgresql
```

you must set it in nuxeo.conf for `nuxeo.template` property

```
nuxeo.templates=postgresql,custom
```

Moreover, the 'database' template does not depend on 'default' template any more, but on 'common' template. So you may have to update your nuxeo.defaults if you previously included 'default'.

On this page

HTTPS

If you're getting the following error at startup: "Expression nuxeo.server.https.keystoreFile is undefined on line 101, column 32 in server.xml.nxftl" (complete stack trace available at NXP-9874), you should change the nuxeo.conf file properties as more properties are now require for HTTPS configuration. If you do not need HTTPS, do not fill any of the https properties. Otherwise, make sure you fill all the properties the startup complains about.

The HTTPS properties to define are:

- `nuxeo.server.https.port=443`
- `nuxeo.server.https.keystoreFile=/path/to/keystore`
- `nuxeo.server.https.keystorePass=password`
- `nuxeo.url`=https://localhost/nuxeo (use your public HTTPS URL here)

The keystore is a file where the JVM will store certificates used and trusted by the HTTPS protocol. It has to be set up using Java's keytool comm and.

Workflow

The workflow system previously used (based on jBPM) has been moved to an addon (Nuxeo jBPM) and is not present anymore in the default Nuxeo distribution. This old workflow system has been replaced by a new, improved one that includes Nuxeo Studio support to design new workflows using a graphical user interface.

Third party libraries upgrades (informative)

- **OpenCMIS** - Nuxeo Platform is now aligned on OpenCMIS 0.7 that comes with support for the CMIS Browser binding (JavaScript compliant API).

Data Migration

Oracle and Clustering

When using Oracle and Nuxeo clustering, you will have to upgrade your cluster tables. See the NXP-9541 Upgrade notes for details.

Custom repository definition

If you have a custom repository descriptor or if you have overridden the default file comming with Nuxeo (`default-repository-config.xml`), be aware that this file now contains 2 contributions :

- `<extension target="org.nuxeo.ecm.core.api.repository.RepositoryManager" point="repositories">`
- `<extension target="org.nuxeo.ecm.core.repository.RepositoryService" point="repository">`

If you have a customized file comming from an earlier version of Nuxeo, the first contribution will be missing (it was previously in nxserver/config/platform-config.xml).

So you will need to add the repository declaration : either in a separated file or directly inside the `default-repository-config.xml`

```
<extension target="org.nuxeo.ecm.core.api.repository.RepositoryManager"
point="repositories">
  <repository name="default" label="Default Repository" />
</extension>
```

NB : If this contribution is missing, the Repository initialization will fail with

```
Caused by: org.nuxeo.ecm.core.api.ClientException: Cannot get repository: default
 at
org.nuxeo.ecm.core.api.UnrestrictedSessionRunner.runUnrestricted(UnrestrictedSessionRu
nner.java:137)
 at
org.nuxeo.ecm.core.repository.RepositoryService.initializeRepository(RepositoryService
.java:166)
```

Code migration

Scheduler service

The name of the Scheduler service component has changed from `org.nuxeo.ecm.platform.scheduler.core.service.SchedulerRegistryService` to `org.nuxeo.ecm.core.scheduler.SchedulerService.`

The descriptor format has not changed so migrating should be as easy as changing extension point usages :

```
<?xml version="1.0"?>
<component name="com.example.nuxeo.schedule.monthly_stuff">
  <extension target="org.nuxeo.ecm.core.scheduler.SchedulerService"
      point="schedule">
    ...
  </extension>
</component>
```

Upgrade from 5.4.2 to 5.5

This chapter presents the detailed process to upgrade from Nuxeo 5.4.2 to Nuxeo 5.5. Most of it is useful information you need to have to fully understand what has changed in this release.
You can have a look at this interesting use case of a "Mostly painless Nuxeo upgrade from 5.4.2 to 5.5 under Windows / PostgreSQL"!

Installation & Configuration

Follow Installation Instruction.

Under all OS:

- H2 Embedded database is not supported for data detection.
- After the installation, uncomment the line `custom.target` in your `template/custom/nux`

eo.default and set it to "." to indicate the path of your custom templates.
- "*session.timeout*" has been added to `nuxeo.conf` and can be overridden for defining the web session timeout which is then integrated into the `web.xml` file.

Under Linux:

- The new installation changes the `opt/nuxeo` content location to `/var/lib/nuxeo/server` - templates and conf still in `etc/nuxeo` folder.
- The package Debian autoconfigure
  - detects your data,
  - detects/adds your marketplace addons (including DM,CMF,DAM).

Under Windows:

- Windows installer:
  - detects your data,
  - detects/adds your marketplace addons (including DM,CMF,DAM).

On this page

Packaging

The DM, DAM, SC, and CMF distributions are now available as Marketplace packages.

This new packaging system is used in the Setup Wizard to allow to choose between different profiles at installation time.
You can also use the Admin Center or the nuxeoctl commands to add or remove these packages. For projects having a custom distribution based on one of ours, no problem, we provide presets for automatically transforming the new unique Tomcat distribution into a DM, DAM or CMF.
Also, the "EAR" (zip) assemblies do still exist.

Using the wizard is just an additional option.

Digital Asset & Case Management

For now, it is not possible to install CMF with other packages like DM and DAM because there are some content model incompatibilities.

Distribution

Regarding to custom distributions and related to the new 5.5 packaging (DM, CMF, DAM are now addons), Ant assembly script (`assembly.xml`) has to be modified:

Deploy the Nuxeo CAP distribution (only nuxeo-cap classifier still exists):

```
<!-- Deploy CAP distribution -->
    <unzip dest="${stagedir}">
      <artifact:resolveFile
key="org.nuxeo.ecm.distribution:nuxeo-distribution-tomcat:${nuxeo.version}:zip"
                            classifier="nuxeo-cap" />
    </unzip>

${stagedir}: distribution parent folder
```

Define type distribution:

```
<!-- Set the addon deploying in distribution -->
    <copy file="${app.path}/nxserver/data/installAfterRestart-?.log"
      tofile="${app.path}/nxserver/data/installAfterRestart.log"
      overwrite="true" />

${app.path}: define your distribution path (ie ./stage/nuxeo-custom-server)
?: DM,DAM,CMF,SC
```

Optional: choose the wizard distribution type by setting wizard addon preset (NXP-8031):

```
<!-- Set the wizard.preset by default -->
 <echo file="${app.path}/setupWizardDownloads/packages-default-selection.properties"
          message="preset=nuxeo-?" />

${app.path}: define your distribution path (ie ./stage/nuxeo-custom-tomcat)
?: dm,cmf,dam
```

Third party libraries upgrades

- **GWT** - Nuxeo is now using **GWT** 2.4.0.
- **JAX-WS** - Libraries have been upgraded to 2.2.5 in order to fix some compatibilities issues.
- **OpenCMIS** - Nuxeo Platform is now aligned on **OpenCMIS** 0.6 that comes with experimental support for the CMIS Browser binding (JS compliant API).
- **JEXL** - Location is changed from Nuxeo Runtime to Nuxeo Platform Action.

Data Migration

VCS

Only fews column additions were done between 5.4.2 and 5.5 (no alter). So you can migrate and retrieve all your 5.4.2 data after 5.5 installation.

For relations, these attributes are added to the "relation" schema (NXP-7962):

```
<xs:element name="predicate" type="xs:string" />
<xs:element name="sourceUri" type="xs:string" />
<xs:element name="targetUri" type="xs:string" />
<xs:element name="targetString" type="xs:string" />
```

A new metadata has been added to follow the legacy definition of `dublincore` schema (NXP-7884) :

```
<xs:element name="publisher" type="xs:string"/>
```

A new schema has been added: `task.xsd` (related to the nuxeo-platform-task feature - NXP-7852):

```
<xs:element name="actors" type="nxt:stringList" /> (Task actors list)
<xs:element name="task_variables" type="nxt:task_variables" /> (tasks vars list)
<xs:element name="taskComments" type="nxt:taskComments" /> (Task comments list)
```

(Four new tables due to complex types added.)

Fulltext

Partially missing fulltext index for the title field

Old versions of Nuxeo DM might have document before the introduction of the "fulltext_title" index. This is visible on the 5.5 release thanks to the new search suggestion widget that might be missing some suggestions on old documents.

To update the title fulltext index, just perform the following SQL query on your PostgreSQL server:

```
UPDATE fulltext SET simpletext_title = NX_TO_TSVECTOR("dublincore"."title") FROM
dublincore WHERE "fulltext"."id" = "dublincore"."id";
```

PostgreSQL fulltext phrase search

In Nuxeo 5.5 for PostgreSQL we've added a better way to store fulltext information that enables the use of phrase search. If you want to use phrase search, you should follow the upgrade notes of NXP-5689. If you do this fulltext upgrade, you may want to check the (unsupported for now) nuxeo-reindex-fulltext plugin to get more accurate phrase search results.

If you don't do the upgrade described in NXP-5689, you'll get the following error message:

```
Cannot use phrase search in fulltext compatibilty mode. Please upgrade the fulltext
table: ...
```

Directories

Directories with auto-incremented columns must be upgraded, as the mechanism for auto-increment has been changed to be more robust. Please follow the NXP-7124 upgrade notes if you have auto-incremented columns (there aren't any in a default Nuxeo installation).

Code Migration

5.5 is mainly backward compatible with 5.4.2. If you have any problems, you can contact Nuxeo Support.

Automation

Changes in Nuxeo Automation: there was a Java package renaming from `org.nuxeo.ecm.automation.client.jaxrs.model` to `org.nuxeo.ecm.automation.client.model`.

Nuxeo Theme

Nuxeo Theme service has been extended so that you can now contribute page styles in a plain CSS stylesheet.
The page layouts are still managed by the Theme engine using an XML description, but all CSS information is now externalized to CSS stylesheets that can manage flavors (pretty much as with LessCSS).

- Theme documentation page
- Migrating a custom theme
- Migrating the branding in Studio to follow Nuxeo changes

Tasks

Until 5.5, the Task system was directly bound to JBPM.
Starting with 5.5, a new TaskService is available and uses VCS to store tasks.
This new TaskService is a first step towards the integration of Content Routing as the default Workflow engine in DM.

Migration should be 100% transparent:

- the Task Operations have not changed,
- REST APIs are maintained,
- Tasks created in jBPM and not directly associated to a process will be automatically migrated upon first access,
- jBPM Tasks are still accessible via the new TaskService,
- the jBPM task API is maintained.

Relations

The new default configuration takes care about compatibility so that if you have existing relations in Jena graph you will still be able to transparently access them.

Upgrade from 5.4.1 to 5.4.2 with Oracle

If you were using Nuxeo DM 5.3.2 or later with Oracle and ACL optimizations enabled, you need to update an index before plugging your database to Nuxeo DM 5.4.2.

```
SELECT index_name FROM USER_INDEXES WHERE table_name LIKE 'READ_ACLS';
```

Get the index name, result of the previous query, which should look like "SYS_C00XXXX"

Then run the following statement:

```
ALTER INDEX "SYS_C00XXXX" RENAME TO "ACLR_ACL_ID_IDX";
```

Now you can start Nuxeo DM 5.4.2 which will rename the table READ_ACLS to ACLR and the previous index will be ready to work with this new configuration.

Upgrade from 5.3.2 to 5.4.0

Installation & Configuration

While Nuxeo EP 5.3.2 was based on the JBoss distribution, Nuxeo EP 5.4.0 is based on Tomcat. There is also a distribution based on JBoss 5, though we strongly recommend you to use the Tomcat distribution: http://cdn.nuxeo.com/nuxeo-5.4.0/nuxeo-dm-5.4.0_01-tomcat.zip.

JBoss 5

If using the JBoss distribution, see Upgrade to 5.4 and JBoss 5.

In this section

Data Migration

On JBoss, data used to be in `$NUXEO_HOME/server/default/data/NXRuntime/`, where you can find the `binaries` folder. Now, with the Tomcat distribution, it stands in `$NUXEO_HOME/nxserver/data`.

- If you have moved your data outside Nuxeo using the `nuxeo.data.dir`, as recommended, you need to move the binaries folder one level up and then remove the `NXRuntime` folder.
- If you haven't moved you data outside Nuxeo, you need to move the `binaries` folder from `$NUXEO_HOME/server/default/data/NXRuntime/` to `$NUXEO_HOME/nxserver/data`.

Code Migration

Workflow

The workflow implementation has changed, see From the old workflow system to the new 5.4 workflow system.

From the old workflow system to the new 5.4 workflow system

Even though the 5.4 jBPM service doesn't implement backward compatibility with the old workflow service, is it possible to deploy both the old workflow framework AND the jBPM service so that the migration can be gradual?

It should be possible to use the m3 version of jBPM and still run the new workflow.
If you create an action for the new workflow tab different from the one for the old workflow, you could use both at the same time. (I assume you are not using publishing tab and forum).

However, I think it would be much easier to move your old workflow to the new one. That would be:

- create the new workflow with handler,
- define what variable you need in the new workflow (most probably docId, repo name ....),
- get the variable from the unfinished process instance in the old jBPM table (object are just serialized into it).

See NXP-2850 for technical details.

Upgrade to 5.4 and JBoss 5

During the migration from JBoss 4.2 to 5.1 we had to do some small changes in Nuxeo bundles.

You will have to do the same for your custom bundles.

Nevertheless, the needed changes are very small and it should only take a few minutes (it should be really quick when you don't have 200 bundles and you know what to do).

Impact on sources and resources

Web resources

Bundles contribution resources to the WAR used to have these resources stored in: `src/main/resources/nuxeo.war`.

This does break JBoss 5 deployment, because it tries to deploy the `nuxeo.war` as a nested WAR, but it fails (because the WAR is not complete and because there are several bundles containing the same `nuxeo.war`).

Rather than changing JBoss deployer config we chose to change our packaging to avoid the problem.

The solution is simple: `nuxeo.war` tree should not be at root of the JAR.

Sample directory structure:

Before (Jboss 4.2)	After (Jboss 5.1)
. `-- src `-- main |-- java `-- resources |-- META-INF |-- nuxeo.war | `-- my_page.xhtml `-- OSGI-INF	. `-- src `-- main |-- java `-- resources |-- META-INF |-- OSGI-INF `-- web `-- nuxeo.war `-- my_page.xhtml

In this section

This implies a small change in the deployment-fragment:

Before (Jboss 4.2)	After (Jboss 5.1)

```
<install>
 <unzip from="${bundle.fileName}"
to="/" >
 <include>nuxeo.war/**</include>
 </unzip>
 </install>
```
	```
	<install>
	 <unzip from="${bundle.fileName}"
	to="/" prefix="web">
	 <include>web/nuxeo.war/**</include>
	 </unzip>
	 </install>
	```

If this is not done, or if there is still a "nuxeo.war" directory present in your jar, you will get an exception like:

```
DEPLOYMENTS IN ERROR:
  Deployment "vfsfile:/opt/jboss/server/default/deploy/nuxeo.ear/" is in error due to
the following reason(s):
java.lang.IllegalStateException: jboss.web.deployment:war=/nuxeo is already installed.
```

Module declaration

From within the deployment fragment you can declare contributions to the application.xml.
Be sure that if you declare your module as a EJB module it does really contains EJB3.

```
<extension target="application#MODULE">
 <module>
 <ejb>${bundle.fileName}</ejb>
 </module>
 </extension>
```

Otherwise, you should remove the contribution to `application.xml` (the Java declaration is not needed and will in fact be ignored by the pre-deployer).

Typical deprecated contribution:

```
<extension target="application#MODULE">
 <module>
 <java>${bundle.fileName}</java>
 </module>
 </extension>
```

Web services binding

The previous versions of Nuxeo were using SUN-JAX-WS (Metro) to handle WebService deployment.
In order to avoid having to modify default JBoss 5.1/EAP 5.0.1 config, we now support to deploy on JBoss native stack (JBoss WS).

This involves doing some small changes in the way the WebService are implemented and deployed.
No more EJB3

Because of some limitation of the JBossWS EJB3 deployer we can not deploy WebServices on top of EJB3 if we want to keep endpoint URLs consistent.
So this simply means you should remove the @Stateless annotation in the Beans providing WebService.

Before (Jboss 4.2)	After (Jboss 5)

@Local(WSAuditLocal.class) @Stateless @Remote(WSAudit.class) @WebService(name = "WSAuditInterface", serviceName = "WSAuditService") @SOAPBinding(style = Style.DOCUMENT) public class WSAuditBean extends AbstractNuxeoWebService implements	@Local(WSAuditLocal.class) @Remote(WSAudit.class) @WebService(name = "WSAuditInterface", serviceName = "WSAuditService") @SOAPBinding(style = Style.DOCUMENT) public class WSAuditBean extends AbstractNuxeoWebService implements

SUN-JAX-WS vs JBossWS binding

Both WS framework need to have a declaration for endpoints, but they (of course) don't use the same way to do it.
Basically, SUN-JAX-WS uses a dedicated `sun-jax-ws.xml` file and JBossWS uses the `web.xml`.

The Nuxeo template system are configured so that you can declare both bindings in your bundle and Nuxeo will deploy the right one depending on the target deployment host.

JBossWS binding	**SUN-JAX-WS binding**

`<!- JBossWS Native EndPoint Declaration ->`
`<extension target="web#JBOSSWS">`
`<servlet>`
`<description>NuxeoAudit WS EndPoint</description>`
`<display-name>NuxeoAudit EndPoint</display-name>`
`<servlet-name>NuxeoAuditEndPoint</servlet-name>`
`<servlet-class>org.nuxeo.ecm.platform.audit.ws.WSAuditBean</servlet-class>`
`</servlet>`
`<servlet-mapping>`
`<servlet-name>NuxeoAuditEndPoint</servlet-name>`
`<url-pattern>/webservices/nuxeoaudit</url-pattern>`
`</servlet-mapping>`
`</extension>`
NB: Yes, you declare you Bean as a servlet even if it does not implement the needed interface!
	`<!`~~- SUN JAXWS / Metro EndPoint Declaration -~~`>`
	`<extension target="jaxws#ENDPOINT">`
	`<endpoint name="nuxeoaudit"`
	`implementation="org.nuxeo.ecm.platform.audit.ws.WSAuditBean"`
	`url-pattern="/webservices/nuxeoaudit" />`
	`</extension>`

Requirements declaration

In JBoss deployed, as it was already the case for tomcat deployment, the Require-Bundle is no longer used by the deployed and should be reserved for OSGi deployment.

The Nuxeo-Require should not longer be used either. `Nuxeo-RequiredBy` will be deprecated but should still be use to override XHTML pages.

In the JBoss 5 deployment (as it was the case in Tomcat), all the Nuxeo bundles are put in the classloader from the start, so there is no risk of ClassNotFound during the loading.

The only dependencies you have to worry about are Runtime dependencies (server availability, contribution override ...): all this should be managed by using the `<require>` tag in the XML contribution.

NB: the `Require-Bundle` is still used when Nuxeo is deployed on an OSGi container, because in this case the class loading issues remains.

Other checks you may want to do

`ejb-jar.xml` files

Unless you know what you are doing, you should remove any `ejb-jar.xml` file from the META-INF folder.
Servlets and filters initialization

Servlets and Filters should not assume that when activated (call to init method by the servlet container) Nuxeo Runtime is ready. It won't be always the case.

So if you need to do some Nuxeo Runtime calls at init time, you should rely on the Framework initialization event, rather on the servlet container init.
EJB3 declaration

JBoss 5 is more strict on the JEE spec, so your EJB3 can not use the same Java interface for @Local and @Remote
Web.xml ordering

JBoss 5 validate the `web.xml` against the DTD and checks order on the tags.

The Nuxeo template is OK and respect the standard, so if you contribute your filters in the rights section and the servlets in the right sections there should be no problem.

But be aware that if you took some shortcuts and contributed several kind of objects (Filters, Servlets, Context-params) in side the same slot, it may have worked OK with JBoss 4 but it won't with JBoss 5.
CoreSession usage

In JBoss environment, CoreSession is delivred via DocumentManagerBean that is a Stateful Session Bean.

JEE spec does not allow concurrent calls on a SFSB : this applies to both JBoss 4 and JBoss 5.

But in the case of JBoss 5 the check is more strict and also prohibits reentrant calls from the same thread.

The typical use case if you create a DocumenModel and you have a Listener that will run and use the core session to do some work.

In JBoss 4 it runs without problem but in JBoss 5 this is detected as a concurrent call.

DocumentManagerBean and DocumentModel implementations have been modified to avoid that, but there are still cases where you may have the problem.

Typical unsafe code is taking the CoreSession via:

```
CoreInstance.getInstance().getSession(doc.getSessionId()).
```

This can be replaced by

```
doc.getCoreSession()
```

that contains the needed logic to detect the reentrant call and return the correct Session (Local or EJB) depending on the context.

So if you have errors like this one:

```
no concurrent calls on stateful bean
'jboss.j2ee:service=EJB3,name=DocumentManagerBean' (EJB3 4.3.13)
```

first check that you don't access the CoreSession from inside a Listener using the DocumentModel sessionId.

Dependencies

Hibernate dependencies in Nuxeo's root `pom.xml` has changed. The core artifact for Hibernate is named `hibernate-core` now instead of `hibernate`. If you were using this dependency, you need to change from:

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate</artifactId>
</dependency>
```

to:

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
</dependency>
```

Impact on the Packaging

Templates

Following structure changes in `nuxeo.ear`, templates structure has changed a little.

If you override a template, check the directories. For instance, here are the `default` template changes:

Before (Jboss 4.2)	After (Jboss 5.1)
. |-- config | |-- default-repository-config.xml | `-- sql.properties |-- datasources | |-- default-repository-ds.xml | `-- unified-nuxeo-ds.xml `-- nuxeo.defaults	. |-- nuxeo-ds.xml |-- nuxeo.defaults `-- nuxeo.ear |-- META-INF | `-- default-repository-ds.xml `-- config |-- default-repository-config.xml `-- sql.properties

As defined in `nuxeo.defaults` ("`default.target=server/default/deploy`"), the target directory of this template is now `server/default/deploy` instead of `server/default/deploy/nuxeo.ear`.

If you defined your own template, you only have to follow the `nuxeo.ear` structure:

- `default-repository-ds.xml` has moved from `nuxeo.ear/datasources/` to `nuxeo.ear/META-INF/`
- `unified-nuxeo-ds.xml` has moved to `nuxeo-ds.xml` and its content now includes mbeans declarations
- `nuxeo.ear/system/` has been renamed to `nuxeo.ear/bundles/`

Assemblies

Deprecated compliance artifacts with old resources were removed.

`nuxeo-platform-ear` and `nuxeo-distribution-dm` now only contain:

nuxeo.ear/
|-- bundles
`-- lib

All resources files are generated from templates directories.

Also, the default EAR archives contain in `lib/` all third-party libraries from the dependency tree (only duplicates are removed). The filtering of provided libraries is done when building the server distribution (JBoss, Tomcat, ...).

Upgrade from 5.3.1 to 5.3.2

Code migration

5.3.2 is fully backward compatible with 5.3.1 (no compat package is needed).

So, you should have no issues with running your custom code against 5.3.2. If you have any problems, you can contact Nuxeo Support.

Packaging

The packaging system is basically the same as the one used in 5.3.1.

The only change that may have an impact involves resources that are now managed by the new template system.

This means that resources are no longer embedded inside the EAR but handled in a separated templates directory.

This makes changing configurations easier (like switching from H2 to PostgreSQL) and will also allow for upgrades without having to redo all custom system configurations.

The documentation has been updated accordingly:

- Installation Guide
- Description of the new configuration system

Data

The only changes done between 5.3.1 and 5.3.2 are the way tags are stored.

Because the Tag Service is now directly part of VCS, some small changes have been done.

Nevertheless, migration should be automatic and transparent.

If you have any problems, you can contact Nuxeo Support.

Configuration

We have changed the way Nuxeo starts OpenOffice.

This is an intermediate solution before we upgrade to JODConverter 3.

The new OOolauncher (that replaces OOodeamon) should:

- be more stable,
- be easier to set up (removed the dependencies on JNI UNO libs).

Upgrade from 5.3.1 with MySQL to 5.3.2

Why a migration script is needed

A database structure change was introduced with Nuxeo 5.3.2 to fix query with operator IN (for more details, see https://jira.nuxeo.com/browse/NXP-5183).

Below is an example of structure change:

Sample structure in Nuxeo DM <= 5.3.1

```
CREATE TABLE `common` (
 `id` varchar(36) NOT NULL,
 `icon` varchar(4000) DEFAULT NULL,
 `icon-expanded` varchar(4000) DEFAULT NULL,
 `size` bigint(20) DEFAULT NULL,
 PRIMARY KEY (`id`),
 KEY `common_id_hierarchy_fk` (`id`),
 CONSTRAINT `common_id_hierarchy_fk` FOREIGN KEY (`id`) REFERENCES `hierarchy` (`id`)
ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Sample structure in Nuxeo DM >= 5.3.2

```
CREATE TABLE `common` (
 `id` varchar(36) CHARACTER SET latin1 COLLATE latin1_bin NOT NULL DEFAULT '',
 `icon` varchar(4000) DEFAULT NULL,
 `icon-expanded` varchar(4000) DEFAULT NULL,
 `size` bigint(20) DEFAULT NULL,
 PRIMARY KEY (`id`),
 KEY `common_id_hierarchy_fk` (`id`),
 CONSTRAINT `common_id_hierarchy_fk` FOREIGN KEY (`id`) REFERENCES `hierarchy` (`id`)
ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

See the part `CHARACTER SET latin1COLLATE latin1_bin` that was added for the column `id`.

As a consequence, the columns (mainly id) used for foreign key don't have the same definition and it is no more possible to add new constraints, needed by the features of Nuxeo DM. trying to upgrade to Nuxeo DM 5.4.x will raise this error

```
java.sql.SQLException: Can't create table 'nuxeodb.#sql-5f31_37d' (errno: 150)
```

Steps for the migration

You need to follow the steps below to migrate your database structure:

- download the script upgradeMySQL.sh attached to this page

- edit these properties in the file upgradeMySQL.sh

```
DB_HOST=localhost
DB_PORT=3306
DB_NAME=nuxeo
DB_USER=user
DB_PWD=password
```

- change permission for the script to run it

```
$ chmod u+x upgradeMySQL.sh
```

- launch the script

```
$ ./updateMySQL.sh
```

- if everything is fine, you'll have a message to confirm the upgrade was done

```
Database structure upgraded successfully
```

Upgrade to Nuxeo DM 5.4.1

Now your database is upgraded, you can test it against Nuxeo DM 5.4.1
Once you've downloaded and unzipped it,

- start it without editing anything: it's needed because you'll need to get the lastest hot fixes to make it work

```
$ cd $NUXEO_HOME/bin
$ ./nuxeoctl start
```

- follow the wizard steps and register on Nuxeo Connect (needed to download Hot Fixes) if you don't have already an account
- restart
- log in using the default credentials
- navigate to Nuxeo Admin Center > Update Center > Software updates tab
- download and install all availables hot fixes (actually at least the three first ones)
- stop the server

```
$ ./nuxeoctl stop
```

- edit the configuration file `nuxeo.conf`and set the database parameters

```
nuxeo.templates=mysql
nuxeo.db.name=nuxeodb
nuxeo.db.user=user
nuxeo.db.password=password
nuxeo.db.host=localhost
nuxeo.db.port=3306
```

- restart the server

```
$ ./nuxeoctl start
```

- contemplate that all data are present 🙂

Upgrade from 5.3.0 to 5.3.1

Nuxeo DM 5.3.1 is fully backward compatible with Nuxeo DM 5.3.0 GA, hence upgrade is painless and requires no data migration or code change.

Follow these steps to upgrade:

1. Get the differences between a vanilla Nuxeo DM 5.3.0 and your custom Nuxeo

If you have specific configuration, you'll need to know what files were changed, in order to apply them onto the default one. Here are the folders and files you have to watch:

- `$JBOSS/server/default/deploy/nuxeo.ear/config/` => main configuration elements of Nuxeo
- `$JBOSS/server/default/deploy/`nuxeo.ear/datasources/ => configuration of data sources
- `$JBOSS/server/default/conf/` => specific configuration of Jboss
- `$JBOSS/server/default/lib/` => specific libraries used by your project (JDBC drivers for instance)
- `$JBOSS/server/default/deploy/mail-service.xml` => configuration of the mail service
- `$JBOSS/server/default/deploy/nuxeo.ear/OSGI-INF/templates/web.xml`

2. Backup your data

- Follow this documentation

3. Apply the differences

- From the differences you got at step 1, apply them on the Nuxeo DM 5.3.1 you've downloaded
- Copy your specific plugins into `$JBOSS/server/default/deploy/nuxeo.ear/plugins/`

4. Restore data

- Copy the data folder (server/default/data) to Nuxeo DM 5.3.1

Upgrades notes

Groups stored in SQL directory

If you were using groups stored in the SQL directory, you have to consider that the "group2group" table must be fixed as its columns were inverted. childGroupId should be populate with the content of parentGroupId and vice-versa. It is related to NXP-4401

Before applying the command below, you have to check that your SQL configuration has changed. This will be the case if you get the new **default-sql-directories-bundle.xml** or if your patch doesn't change the **tableReference** attribute defined in the new **default-sql-directories-bundle.xml** The attribute should look like:

```
<tableReference field="subGroups" directory="groupDirectory"
  table="group2group" sourceColumn="parentGroupId"
  targetColumn="childGroupId" schema="group2group" />
```

In that case, run the following query for PostgreSQL to update the table:

```
UPDATE group2group SET "childGroupId" = "parentGroupId", "parentGroupId" =
"childGroupId";
```

OpenSocial

- The `opensocial.properties` file format has been changed in 5.3.1, so you may need to use the one provided in 5.3.1 rather that trying to upgrade the one used in 5.3.
- For a dashboard initialized in 5.3.0, existing OpenSocial gadgets need to be migrated: see NXP-4923 for script and procedure.

Indexing

There are no impacting changes on the storage structure.
If you want to leverage the new default indexing configuration (separated full-text index for title and description), you will have to update your repository configuration (or use the one provided with 5.3.1) and build the new indexes.

For developers

If you use a custom Nuxeo assembly to package your Nuxeo distribution with your plugins, you will need to modify your existing assembly.

The new `nuxeo-distribution` system is simpler to configure and extend than the previous one.

See here and here for more details.

Upgrade from 5.1.6 with JCR + PostgreSQL to 5.2.0

This article will help you to migrate your data from Nuxeo 5.1.6 to Nuxeo 5.2 in the case you are using JackRabbit with PostgreSQL as backend.

We assume that your Nuxeo 5.1.6 is installed in $JBOSS_516 directory and Nuxeo 5.2 in $JBOSS_52 and you have well configured your Nuxeo 5.2 to work with Jackrabbit/PSQL. Otherwise, let's see this article.

The steps to migrate are:

- Start an empty nuxeo 5.2 configured in JCR
  - Customize a nuxeo 5.2 JCR with an **emtpy** database, created for the occasion.
  - Start nuxeo 5.2 and log in
  - shutdown nuxeo 5.2
- copy the file $JBOSS_52/server/default/data/NXRuntime/repos/default/repository/nodetypes/custom_nodetypes.xml and keep it in a temporary location
- keep either the directory $JBOSS_52/server/default/data/NXRuntime/repos/default/repository/namespaces/
- remove $JBOSS_52/server/default/data
- copy the data folder from $JBOSS_516/server/default/data to $JBOSS_52/server/default/data
- copy the custom_nodetypes.xml file you kept to $JBOSS_52/server/default/data/NXRuntime/repos/default/repository/nodetypes/
- change searchIndex class to org.nuxeo.ecm.core.repository.jcr.jackrabbit.SearchIndex in $JBOSS_52/server/default/data/NXRuntime/repos/default/workspaces/default/workspace.xml
- remove the $JBOSS_52/server/default/data/NXRuntime/repos/default/workspaces/default/index folder to force JackRabbit to rebuild the indexes
- update discriminator column in NXP_LOGS table to allow this value to be null
  - alter table NXP_LOGS alter discriminator DROP not null
- Here is the tricky part, customize the ns_idx.properties in the directory namespaces that you kept:

```
Compare the file
$JBOSS_52/server/default/data/NXRuntime/repos/default/repository/namespaces/ns_idx.pro
perties  with the one you kept from the namespaces directory.
They contain uri and an identifier, example :
http://www.nuxeo.org/ecm/schemas/common/=19
Each identifier is unique!
You need to adapt the ns_idx.properties keeped in order that each uri keep is old
identifier unchanged .
```

simple example :

$JBOSS_52/server/default/data/NXRuntime/repos/default/repository/namespaces/ns_idx.properties

```
http://www.nuxeo.org/ecm/schemas/common/=21
http://www.nuxeo.org/ecm/schemas/dublincore/=18
http://project.nuxeo.org/schemas/webengine/site/blog/post=19
```

$JBOSS_516/server/default/data/NXRuntime/repos/default/repository/namespaces/ns_idx.properties

```
http://www.nuxeo.org/ecm/schemas/common/=19
http://www.nuxeo.org/ecm/schemas/dublincore/=18
```

As you see in the 516 file, http://www.nuxeo.org/ecm/schemas/common/ was identified by 19, so we need to keep this identifier, but http://project.nuxeo.org/schemas/webengine/site/blog/post is already identified by 19 so we will just switch the two identifiers.
Here is the new file :

```
http://www.nuxeo.org/ecm/schemas/common/=19
http://www.nuxeo.org/ecm/schemas/dublincore/=18
http://project.nuxeo.org/schemas/webengine/site/blog/post=21
```

- remove the directory $JBOSS_52/server/default/data/NXRuntime/repos/default/repository/namespaces/
- copy your namespace directory (With the customized ns_idx.properties) in $JBOSS_52/server/default/data/NXRuntime/repos/default/repository/namespaces/
- finally adapt nuxeo 5.2 to use the 5.1.6 database.

Just for information, below are the changes you can make manually to update your custom_nodetypes.xml

Two main problems occurs are present in the node type definitions from Nuxeo 5.1.6:

- the whole versioning features are not working : no document modification, no version increase, no reading of the previous versions, ...: this is due to the fact that ecm:version and ecm:versionHistory are not mixin type any more. Manually you can change these nodes and chose isMixin="true" to isMixin="false"
- some document definitions have changed :
    - Workspace type has two new supertypes: ecmst:publish_ergo and ecmst:webcontainer
    - Forum, Thread and post types use now "ecmdt:Document" as supertype instead of "ecmnt:document"
    - WikiPage and BlogPost types use "ecmmix:versionable" as supertype instead of "mix:versionable"

Editing the custom_nodetypes file is not easy because you have to format this file (tidy -xml ...) to edit it. So we recommend to replace the old custom_nodetypes.xml by the new one, generated from a fresh Nuxeo 5.2 installation.

Upgrade from 5.1.2 to 5.1.3

Follow Upgrade Nuxeo and apply the following procedure _*before*_ starting Nuxeo.

While upgrading from 5.1.2 to 5.1.3, you may have to manage with a Blob format issue : that means to patch $JBOSS/server/default/data/NXRuntime/repos/default/repository/nodetypes/custom_nodetypes.xml
Take this file, format it with tidy (tidy -wrap 999 -indent -xml) and apply this patch (manually as it can't guarantee any line numbers; add the lines beginning with a "+" if not already present):

```
+  <nodeType hasOrderableChildNodes="false" isMixin="true" name="ecmmix:content"
primaryItemName="">
+    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="digest" onParentVersion="COPY" protected="false" requiredType="String" />
+    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="length" onParentVersion="COPY" protected="false" requiredType="Long" />
+    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="filename" onParentVersion="COPY" protected="false" requiredType="String" />
+  </nodeType>
   <nodeType hasOrderableChildNodes="false" isMixin="false" name="ecmft:content"
primaryItemName="">
     <supertypes>
       <supertype>ecmnt:property</supertype>
+      <supertype>ecmmix:content</supertype>
+      <supertype>nt:resource</supertype>
     </supertypes>
    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="mime-type" onParentVersion="COPY" protected="false" requiredType="String" />
    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="data" onParentVersion="COPY" protected="false" requiredType="Binary" />
    <propertyDefinition autoCreated="false" mandatory="false" multiple="false"
name="encoding" onParentVersion="COPY" protected="false" requiredType="String" />
   </nodeType>
```

Then, you need to re-index your data. Using nuxeo-shell (versus web function in advanced search) is recommended.

Another solution (than patching custom_nodetypes.xml file) is to export then re-import data before and after the upgrade (using nuxeo-shell too); but this method will make you loose versioning information.

How to Replicate the Nuxeo Repository

The system replication feature aims at cloning entire collections of data existing in a Nuxeo system. Consequently the clone is importable in another system leading to a complete replication of the system.

Such feature is obviously an important gain because it allows:

- A complete backup of the system,
- A complete data migration,
- Replication of complex systems.

The feature is in fact an export – import tool. Three projects are designed to accomplish the objectives:

- A common module,
- An export module and
- A import module.

In this section

The export module allows migration from older storages to current supported Nuxeo deployment.

In this current stage, only the documentary base is replicated. The relations, vocabularies, users replications are left for later implementation. That because:

- Relations are stored usually in Jena storage: this can be moved as it is. As long as replication preserves the document ID the relations can be moved independently.
- Vocabularies are also stored in directory structures which can be moved independently.
- Users and groups are usually stored outside the Nuxeo repository (LDAP) which supposedly remain the same. If it is not the case, the customer needs to previously ensure the same user / groups structure is in place.

Another important constraint is that the import on H2 database could crash.

The resume feature is not yet implemented.

The selection of a part of repository to be exported, respectively imported is not yet implemented.

More can be found at http://doc.nuxeo.org/5.2/books/nuxeo-book/html-single/#admin-replication.

The export and import services are made available either though JSF UI or through a MBean visible in JMX console. The UI is available only if the web component is also deployed. The feature code is currently located in addon project nuxeo-platform-replication (as also nuxeo-platform-importer feature, which is used inside replication). In order to have it deployed the following JARs needs to be deployed:

- Export:
  - `nuxeo-platform-replication-common-api-$version.jar`
  - `nuxeo-platform-replication-common-core-$version.jar`
  - `nuxeo-platform-replication-exporter-api-$version.jar`
  - `nuxeo-platform-replication-exporter-core-$version.jar`
  - `nuxeo-platform-replication-exporter-web-$version.jar`
  - `nuxeo-platform-replication-exporter-mbean-5.1.6.sar (only for 5.1.6) with applicable versions: 5.1.6, 5.3.0, 5.3.2, 5.4.2`
- Import:
  - `nuxeo-platform-importer-core-$version.jar`
  - `nuxeo-platform-replication-common-api-$version.jar`
  - `nuxeo-platform-replication-common-core-$version.jar`
  - `nuxeo-platform-replication-importer-api-$version.jar`
  - `nuxeo-platform-replication-importer-core-$version.jar`
  - `nuxeo-platform-replication-importer-web-$version.jar with applicable versions: 5.2.1-SNAPSHOT, 5.3.0, 5.3.2, 5.4.2`

Export Instructions – JSF UI

The link “Export” is present in the top list of actions. The page allows selecting the destination of the archive. The exported artifacts are stored here, so enough space must be ensured.

Once the export launched (clicking once on the link “Export”) the page displays the progress, updating the number of documents exported in top of the page. At the end of export, status “Done” is displayed.

Please be patient and don't press twice the link.

Also, the status and information about the export can be seen in the server log (see below).

Export Instructions – JMX Console MBean

The beans interface can be found in JMX console, under the “nx” section, as service “Exporter” on 5.1.6, respectively “ExporterService” on 5.2. The method “export” requires two parameters: the repository name (usually “default”) and the path to save the archive (the third boolean one is not effective).

For the results, watch the server logs (see below).

Import instructions – JSF UI

The link “Import” is present in the top list of actions. The page allows selecting the source of the archive.

Once the import launched (pressing once on the link “Import”) the page displays the progress, updating the number of documents imported in top of the page. At the end of import, status “Done” is displayed.

Please log out and log in in order to refresh the view.

Please be patient and don't press twice the link.

Also, the status and information about the import can be seen in the server log.

Import Instructions – JMX Console MBean

Release 5.2 benefits the existence of MBeans. The beans interface can be found in JMX console, under the “nx” section, as service

“ImporterService”. The method “importDocuments” requires one parameter: the path to save the archive.
For the results, watch the server logs.

The export or import results are best viewed in the logging system, as configured in JBoss. Attention: the level of logging for replication tool need to be at least INFO to see the summary. The summary contains

- The number of documents attempted to be exported,
- The type of errors encountered, each one with,
- A short description,
- The list of documents in fault.
  An example of how this summary looks:

```
15:41:08,200 INFO [ExporterReporter] Summary of export action
15:41:08,200 INFO [ExporterReporter] 108005 documents attempted to export
15:41:08,200 INFO [ExporterReporter] time elapsed: 45 minutes, 48 seconds and
38 milliseconds, velocity: 39.30
15:41:08,200 INFO [ExporterReporter]   2 documents are missing a version.
15:41:08,200 INFO [ExporterReporter]   They are still available for import
with no versions attached.
15:41:08,200 INFO [ExporterReporter]     version <unknown> for document
/home/user/export/Documentary Base/Usual
documents/default-domain/sections/sectiunea/notade_1257254138886
15:41:08,200 INFO [ExporterReporter]     version <unknown> for document
/home/user/export/Documentary Base/Usual
documents/default-domain/sections/sectiunea/notades_1257254164645
15:41:08,200 INFO [ExporterReporter]   1 documents are missing a blob file.
15:41:08,200 INFO [ExporterReporter]   Still they are available for import
with a fake blob file instead.
15:41:08,200 INFO [ExporterReporter]     95d1c8b2.blob for document
/home/user/export/Documentary Base/Usual
documents/default-domain/workspaces/deee/filefile
```

  or if no error was recorded the message below is displayed:

```
Operation completed with no errors recorded.
```

Exporting

The following errors are possible:

- unknown system error, with the message

```
<# of documents> documents yields unexpected error.
Their status is undefined from the exporter perspective.
```

  A number of documents exported ended in error. It can't be told if they are available for import or not. Logs need to be consulted for more details.
- document structure is corrupted, with the message

```
<# of documents> documents are compromised.
They couldn't be exported. Check log for more details.
```

  A number of documents have the XML structure corrupted and can't be recovered. These documents can't be exported / imported.
- missing children / versions, with the message

```
for <# of documents> documents children are not available.
The children couldn't be read: they are not listed nor exported.
```

respectively

```
for <# of documents> documents versions are not available.
The versions couldn't be read: they are not listed nor exported.
```

These documents were searched for children / versions, but due to errors reading, the lists couldn't be retrieved. Obviously, the children / versions (is any) are not registered for export and not attempted to be exported. Well, at least not from this try, but maybe as version published, or some other situations.

- a particular version is missing, with the message

```
<# of documents> documents are missing a version.
They are still available for import with no versions attached.
```

Usually, for various documents (proxy or usual documents) a version is registered but can't be found. The export treats the document as no version attached (the import will act accordingly).

- head of a version is missing, with the message

```
<# of documents> versions are orphans.
They are still available for import with no live document attached.
```

The versions are exported, but no live document is found. Well, the versionned documents will be available (as published versions for instance) but with no living head.

- blobs are missing, with the message

```
<# of documents> documents are missing a blob file.
Still they are available for import with a fake blob file instead.
```

The documents are exported with a fake blob file instead the expected one. The fake blob is a text file containing "The original blob could not be found, this is a fake replacement one."

> The mime type and file name are not modified.

Importing

The following errors are possible:

- unknown system error, with the message

```
<# of documents> documents yields unexpected error.
Their status is undefined from the importer perspective.
```

A number of documents imported ended in error. It can't be told if they are available for import or not. Logs needs to be consulted for more details.

- document structure is corrupted, with the message

```
<# of documents> documents are compromised.
They are imported but as empty documents - including the title.
```

A number of documents have the XML structure corrupted and can't be recovered. These documents were imported, but with no schemes or data inside.
- repository import of a document failed, with the message

```
<# of documents> documents failed to be cloned in repository.
They couldn't be imported. Check log for more details.
```

The atomic operation of importing the document in repository failed (because wrong configuration of repository / available types; because the exported document structure is corrupted or obsolete, etc). The import can't be performed.
- applying the custom schema change failed, with the message

```
<# of documents> documents custom schema update failed.
The documents are imported as they are, without any custom change.
```

The custom defined schema change was applied but it threw exception. The changes were discarded and the initial document structure is imported instead.
- document type denied to be imported, with the message

```
<# of documents> documents were rejected based on the type selection.
The documents are not imported.
```

The documents were not imported as they are denied by type choice to be imported.
- the ACL couldn't be applied on a document, with the message

```
<# of documents> documents failure to update the ACL system.
The documents are imported and preserved with default security rights.
```

The documents were imported with no local ACL changes.

1. The export facility was tested and proved working fine on 5.1.6 and 5.2 servers with various backend storages (JCR, H2, PostgreSQL). The import was tested on 5.2 only with PostgreSQL backend. It is known that H2 doesn't support the import. The error thrown is timeout trying to lock a table. Nor JCR supports the import, as the node ID can't be created voluntarily in JCR system.
2. When exporting proxies, the targeted documents are also exported, increasing redundantly the size of archive: http://jira.nuxeo.org/browse/NXP-3825.
3. Sometimes the UI could crash after or during the import, because the documents are changed massively in repository. Just log in again.
4. The visual reports are sometime broken. The data recorded in log are the one to be considered as real.

Nuxeo Security Hotfixes

Nuxeo provides security hot fixes to correct flaws or other security issues. It is highly recommended to install the following packages.

Nuxeo Security Hotfix 01

This package fixes the RichFaces CVE-2013-2165 flaw.

JBoss RichFaces has a known flaw related to deserialization: https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2013-2165

Details of the patch are here: http://www.bleathem.ca/blog/2013/07/richfaces-CVE-2013-2165.html

Nuxeo has assigned NXBT-661 and NXP-13112 to apply the official patch.

Affected Versions

This security issue affects Nuxeo Platform 5.6 and 5.8. Nuxeo provides hotfixes that automatically include this security fix: Nuxeo 5.6.0-HF27 and 5.8.0-HF-01.

Any earlier version of these two target platforms needs this security hot fix. Any later version will include the fix.

Installing Nuxeo Security Hotfix 1

Installation from the Marketplace

On a registered instance, you can install this security hot fix from the Admin Center of your Nuxeo instance.

1. Go to **Update Center** > **Nuxeo software update.**
2. Install **nuxeo-security-HF01-1.0.0** like any other hot fix.
3. If required, restart your server.

On an instance that is not registered:

1. Go to your `$NUXEO_HOME/bin` directory.
2. Run `nuxeoctl mp-install nuxeo-security-HF01`.

Your Nuxeo server(s) need to have Internet access.

Manual installation

The security hot fix provides 2 RichFaces patched JARs:

- richfaces-impl-3.3.1.GA-NX9.01.jar
- richfaces-ui-3.3.1.GA-NX9.01.jar

You can manually update these JARs with the following steps:

1. Go to `$NUXEO_HOME/nxserver/lib`.
2. Locate `richfaces-impl-3.3.1.GA-NXxx.jar` and `richfaces-ui-3.3.1.GA-NXxx.jar`

   If xx is above 9.01, your JARs already include the security fix, there is nothing to do.
3. Remove these two JARs and replace them with the ones mentioned above.

Credits

We'd like to thank Arun Neelicattu and David Jorm from Red Hat for reporting this issue.

In this section

Logs Analysis

Logging Configuration

Nuxeo logging is compliant with common Java logging frameworks Log4J, SLF4J and JUL.

Logging in Nuxeo is configured through a Log4J XML file:

- `$NUXEO_HOME/lib/log4j.xml` (for Tomcat),
- `$NUXEO_HOME/server/default/conf/jboss-log4j.xml` (for Nuxeo 5.2+ with JBoss).

Editing that file, you can set the logging rules (log level per file and component, files rotation, ...).

Log4J log levels are: ERROR, WARN, INFO, DEBUG and TRACE.

You can increase or decrease the logs for specific services. Here are some useful informations:

- `org.nuxeo.runtime.deployment.preprocessor.DeploymentPreprocessor` logs the pre-processing order,
- `org.nuxeo.osgi.application.loader.FrameworkLoader` writes the configuration summary,
- `org.nuxeo.runtime.osgi.OSGiRuntimeService` provides the final startup summary.

In this section

Related pages:

- Log4J
- How to change the JBoss log files rotation
- Where are the log and configuration files in Windows?

Log Files

The log files location depends on `nuxeo.log.dir` parameter. By default, Nuxeo writes five log files:

- nuxeoctl.log
  Log activity from NuxeoCtl.
- console.log
  Log output written to the console.
- server.log
  Server logs.
- nuxeo-error.log
  Gather errors raised to the user.
- stderr.log
  Gather errors written to the console.

Tomcat Specific

The following information is for debug purpose, it may have impacts on performance and logging files size

`$NUXEO_HOME/conf/logging.properties` has no effect; you can safely remove it.

Because Tomcat uses `java.util.logging` of which Log4J is not aware, Tomcat logs (`org.apache.catalina.*`) have to be redirected to Apache Commons Logging (compliant with Log4J).

Nuxeo provides a bridge for that redirection (in org.nuxeo.common.logging.JavaUtilLoggingHelper) with a default threshold set at INFO level. You can customize it by adding into `nuxeo.conf`:

```
JAVA_OPTS=$JAVA_OPTS -Dorg.nuxeo.runtime.redirectJUL.threshold=CONFIG
```

But that redirection is only available after Nuxeo Framework initialization whereas most of the startup logs from `org.apache.catalina` happen before (when the redirection is not yet active).

Note also that FINER and FINEST logs will be ignored anyway.

Common Logs

Here are descriptions of the most common logs coming from Nuxeo or its third-parties. The list is not exhaustive but aims at helping to understand the most common logs you may encounter while using the Nuxeo platform. If you don't find what you need here, you can check answers.nuxeo.com if someone already asked about it. You can also google third parties logs if they are not in this list.

Level	Source Message	Cause
	`Cannot set databaseTransactionEnabled attribute to false for model publication, other models already exist with value true`	Not a problem. The databaseTransactionEnabled is a Jena issue.
	`Oracle 11g is not yet fully supported; using 10g dialect`	Not a problem. Because not using the latest Hibernate version. It's not a problem either given the SQL generated by Nuxeo.

WARN	`org.hibernate.ejb.Ejb3Configuration (org.hibernate:hibernate-entitymanager)`
	
	`Overriding hibernate.transaction.factory_class is dangerous, this might break the EJB3 specification implementation`
		Not a problem.
		
		Generated by Hibernate and can be ignored. This is voluntarily done and is under control.

WARN	`org.jboss.seam.core.Init (org.jboss.seam:jboss-seam)`
	
	`The built-in interceptor org.jboss.seam.persistence.EntityManagerProxyInterceptor is missing. This application may not function as expected`
	
	Similar messages with: `org.jboss.seam.bpm.BusinessProcessInterceptor`, `org.jboss.seam.core.BijectionInterceptor`, `org.jboss.seam.webservice.WSSecurityInterceptor`, `org.jboss.seam.security.SecurityInterceptor`.
		Not a problem.
		
		Some unneeded built-in interceptors are removed for performance reasons.

WARN	`org.jboss.seam.Component (org.jboss.seam:jboss-seam)`
	
	`Component class should be serializable: org.jboss.seam.ui.facelet.mockHttpSession`
		Third-party issue.
		
		Seam provides a non-serializable component and then warns about it. Even if we don't use it, it's still part of the default Seam JARs.

WARN	`org.nuxeo.ecm.core.search.api.client.querymodel.QueryModelService (org.nuxeo.ecm.platform:nuxeo-platform-search-api)`
	
	`Query models are deprecated as of Nuxeo 5.4 and will be removed for Nuxeo 6.0: the query model '...' should be upgraded to use content views`
		Use of deprecated API. See Javadoc.

WARN	`org.nuxeo.ecm.webapp.pagination.ResultsProviderService (org.nuxeo.ecm.platform:nuxeo-platform-webapp-base)` `Result providers are deprecated as of Nuxeo 5.4 and will be removed for Nuxeo 6.0: the result provider '...' should be upgraded to use content views`	Use of deprecated API. See Javadoc.
WARN	`org.nuxeo.theme.styling.service.ThemeStylingServiceImpl (org.nuxeo.theme:nuxeo-theme-styling)` `Style unknown: helpers`	
WARN	`org.artofsolving.jodconverter.office.OfficeProcess (org.artofsolving.jodconverter:jodconverter-core)` `Restarting OOo after code 81 ...`	See jodconverter issue 84.
WARN	`org.artofsolving.jodconverter.office.StreamGobbler (org.artofsolving.jodconverter:jodconverter-core)` `StreamGobbler: Fontconfig warning: "/usr/lib/libreoffice/share/fonts/truetype/fc_local.conf", line 13: Having multiple <family> in <alias> isn't supported and may not works as expected`	
WARN	`org.jboss.seam.init.Initialization (org.jboss.seam:jboss-seam)` `namespace declared in components.xml does not resolve to a package:`	
WARN	`org.jboss.seam.security.permission.PersistentPermissionResolver (org.jboss.seam:jboss-seam)` `no permission store available - please install a PermissionStore with the name 'org.jboss.seam.security.jpaPermissionStore' if persistent permissions are required.`	
`WARN`	`org.nuxeo.ecm.core.api.TransactionalCoreSessionWrapper (org.nuxeo.ecm.core:nuxeo-core)` `Session invoked in a container without a transaction active: turn on debug logs for more information about the faulty call.`	See Unable to render JIRA issues macro, execution error. upgrade notes.

Error rendering macro 'contentbylabel' : com.atlassian.confluence.macro.params.ParameterException: 'NXDOC5856' is not an existing space's key

Purging Audit Logs (NXP_LOGS)

Depending on usage (lots of updates, lots of workflows, lots of logins, ...), the audit tables (`NXP_LOGS` and related tables) can grow very quickly.

You can configure the audit to filter what must be recorded, but there is no API or UI to do a cleanup inside the audit tables. Actually, this is not something we forgot, we simply considered that it was safer like that: the Audit Service is here to record activity in the platform, it makes sense that a component cannot easily delete its audit trail.

This means that the cleanup should be done at the SQL level. Since the table structure of `NXP_LOGS` is really obvious, it is an easy job for a database administrator to remove old rows based on the `log_event_date` column which contains a timestamp.

If you prefer you can find below the source of a PostgreSQL function that can be used to purge Audit entries older than a given date. You can easily adapt it:

- to change the filtering done on audit record to filter,
- to match the syntax of other databases than PostgreSQL.

nx_audit_purge for PostgreSQL

```
CREATE OR REPLACE FUNCTION nx_audit_purge(olderThan character varying)
 RETURNS int AS
$BODY$
DECLARE
-- INPUT format is 'YYYY-MM-DD'
 maxDate varchar(11) := olderThan;
 nblines int;
 total int;
BEGIN
-- Because nxp_logs_mapextinfos has 2 FK on external tables
-- we must remove records from this table first
-- so we need to store the values in a tmp table before
CREATE TEMP TABLE audit_purge_tmp ON COMMIT DROP AS
 SELECT nxp_logs_mapextinfos.log_fk, nxp_logs_mapextinfos.info_fk
 FROM nxp_logs, nxp_logs_extinfo, nxp_logs_mapextinfos
 WHERE
 nxp_logs.log_event_date < maxDate::date
 AND nxp_logs_mapextinfos.log_fk = nxp_logs.log_id
 AND nxp_logs_mapextinfos.info_fk=nxp_logs_extinfo.log_extinfo_id;
-- CLEANUP on nxp_logs_mapextinfos bridge table first to drop constraints
RAISE INFO 'run cleanup on nxp_logs_mapextinfos (level 2)';
DELETE
 FROM nxp_logs_mapextinfos
 WHERE nxp_logs_mapextinfos.log_fk IN (
 SELECT log_fk FROM audit_purge_tmp);
GET DIAGNOSTICS nblines = ROW_COUNT;
 SELECT nblines INTO total;
RAISE INFO '% lines cleanup on table nxp_logs_mapextinfos' ,nblines;
-- LEVEL 3 cleanup

 RAISE INFO 'run cleanup on nxp_logs_extinfo (level 3)';

 DELETE
 FROM nxp_logs_extinfo
 WHERE nxp_logs_extinfo.log_extinfo_id IN (
 SELECT info_fk FROM audit_purge_tmp);
GET DIAGNOSTICS nblines = ROW_COUNT;
 SELECT nblines+total INTO total;
RAISE INFO '% lines cleanup on table nxp_logs_extinfo' ,nblines;
-- LEVEL1 cleanup
RAISE INFO 'run cleanup on nxp_logs (level 1)';

 DELETE
 FROM nxp_logs
 WHERE nxp_logs.log_id IN (SELECT log_fk FROM audit_purge_tmp);
GET DIAGNOSTICS nblines = ROW_COUNT;
 SELECT nblines+total INTO total;

 RAISE INFO '% lines cleanup on table nxp_logs' ,nblines;
 RAISE INFO '% lines total cleanup ' ,total;

 RETURN total;
END $BODY$
 LANGUAGE plpgsql VOLATILE
 COST 100;
ALTER FUNCTION nx_audit_purge(character varying)
 OWNER TO nuxeo;
```

Here is the same script for SQL Server:

nx_audit_puge for SQL Server

```
ALTER PROCEDURE [dbo].[nx_audit_purge]
  @olderThan varchar(11)
AS
BEGIN
  -- SET NOCOUNT ON added to prevent extra result sets from
  -- interfering with SELECT statements.
  SET NOCOUNT ON;
  -- INPUT format is 'YYYYMMDD'
  DECLARE @maxDate varchar(11) = @olderThan;
  DECLARE @nblines int = 0;
  DECLARE @total int = 0;

  -- Because nxp_logs_mapextinfos has 2 FK on external tables
  -- we must remove records from this table first
  -- so we need to store the values in a tmp table before
  SELECT NXP_LOGS_MAPEXTINFOS.LOG_FK, NXP_LOGS_MAPEXTINFOS.INFO_FK
    INTO #audit_purge_tmp
    FROM NXP_LOGS, NXP_LOGS_EXTINFO, NXP_LOGS_MAPEXTINFOS
    WHERE NXP_LOGS.LOG_EVENT_DATE < convert(datetime,@maxDate,112)
    AND NXP_LOGS_MAPEXTINFOS.LOG_FK = NXP_LOGS.LOG_ID
    AND NXP_LOGS_MAPEXTINFOS.INFO_FK=NXP_LOGS_EXTINFO.LOG_EXTINFO_ID

  -- CLEANUP on nxp_logs_mapextinfos bridge table first to drop constraints
  RAISERROR ('run cleanup on nxp_logs_mapextinfos level 2',0,1);
  DELETE FROM NXP_LOGS_MAPEXTINFOS
    WHERE NXP_LOGS_MAPEXTINFOS.LOG_FK IN (SELECT LOG_FK FROM #audit_purge_tmp);
  SET @nblines = @@ROWCOUNT;
  SET @total = @nblines;
  RAISERROR ('% lines cleanup on table nxp_logs_mapextinfos',0,1,@nblines);

  -- LEVEL 3 cleanup
  RAISERROR ('run cleanup on nxp_logs_extinfo level 3',0,1);
  DELETE FROM NXP_LOGS_EXTINFO
    WHERE NXP_LOGS_EXTINFO.LOG_EXTINFO_ID IN (SELECT INFO_FK FROM #audit_purge_tmp);
  SET @nblines = @@ROWCOUNT;
  SET @total = @nblines+@total;
  RAISERROR ('% lines cleanup on table nxp_logs_extinfo' ,0,1,@nblines);

  -- LEVEL 1 cleanup
  RAISERROR ('run cleanup on nxp_logs level 1',0,1);
  DELETE FROM NXP_LOGS
    WHERE NXP_LOGS.LOG_ID IN (SELECT LOG_FK FROM #audit_purge_tmp);
  SET @nblines = @@ROWCOUNT;
  SET @total = @nblines+@total;

  RAISERROR ('% lines cleanup on table nxp_logs' ,0,1,@nblines);
  RAISERROR ('% lines total cleanup ' ,0,1,@total);
  DROP TABLE #audit_purge_tmp;
  RETURN @total
END
```

Here for Oracle:

nx_audit_puge for Oracle

```
CREATE GLOBAL TEMPORARY TABLE audit_purge_tmp (
  log_fk NUMBER(38),
  info_fk NUMBER(19)
) ON COMMIT DELETE ROWS;

CREATE OR REPLACE PROCEDURE nx_audit_purge(olderThan VARCHAR2)
IS
  -- INPUT format is 'YYYY-MM-DD'
  maxDate VARCHAR2(10) := olderThan;
  nblines PLS_INTEGER;
  total PLS_INTEGER;
BEGIN
  -- Because nxp_logs_mapextinfos has 2 FK on external tables
  -- we must remove records from this table first
  -- so we need to store the values in a tmp table before
  INSERT INTO audit_purge_tmp
    SELECT nxp_logs_mapextinfos.log_fk, nxp_logs_mapextinfos.info_fk
      FROM nxp_logs, nxp_logs_extinfo, nxp_logs_mapextinfos
      WHERE nxp_logs.log_event_date < TO_DATE(maxDate, 'YYYY-MM-DD')
      AND nxp_logs_mapextinfos.log_fk = nxp_logs.log_id
      AND nxp_logs_mapextinfos.info_fk=nxp_logs_extinfo.log_extinfo_id;
  -- CLEANUP on nxp_logs_mapextinfos bridge table first to drop constraints
  dbms_output.put_line('Run cleanup on nxp_logs_mapextinfos (level 2) ...');
  DELETE FROM nxp_logs_mapextinfos
    WHERE nxp_logs_mapextinfos.log_fk IN (SELECT log_fk FROM audit_purge_tmp);
  nblines := SQL%ROWCOUNT;
  total := nblines;
  dbms_output.put_line('Lines cleanup on table nxp_logs_mapextinfos: '|| nblines);
  -- LEVEL 3 cleanup
  dbms_output.put_line('Run cleanup on nxp_logs_extinfo (level 3) ...');
  DELETE FROM nxp_logs_extinfo
    WHERE nxp_logs_extinfo.log_extinfo_id IN (SELECT info_fk FROM audit_purge_tmp);
  nblines := SQL%ROWCOUNT;
  total := total + nblines;
  dbms_output.put_line('Lines cleanup on table nxp_logs_extinfo: ' || nblines);
  -- LEVEL 1 cleanup
  dbms_output.put_line('Run cleanup on nxp_logs (level 1) ...');
  DELETE FROM nxp_logs
    WHERE nxp_logs.log_id IN (SELECT log_fk FROM audit_purge_tmp);
  nblines := SQL%ROWCOUNT;
  total := total + nblines;
  dbms_output.put_line('Lines cleanup on table nxp_logs: ' || nblines);
  dbms_output.put_line('Total lines cleanup: '|| total);
END;
```

Related sections

- Purging Audit Logs (NXP_LOGS) (Nuxeo Installation and Administration - 5.8)
- Configuring Oracle (Nuxeo Installation and Administration - 5.8)
- Transactions and Connections (Nuxeo Installation and Administration - 5.8)
- Directories and Vocabularies (Nuxeo Platform Developer Documentation - 5.8)
- Examples of SQL Generated by VCS (Nuxeo Platform Developer Documentation - 5.8)
- Logs Analysis (Nuxeo Installation and Administration - 5.8)
- Recommended Configurations (Nuxeo Installation and Administration - 5.8)

Remote Monitoring Through a SSH Tunnel

At some time, you may need to monitor your server trough your SSH access. We assume you have a direct connection to your remote server host. If you're using a gateway, this works too, you just have to configure the right ports forwarding.

Here is how to monitor your server through a SSH tunnel:

1. On server host, run `jstatd` with the privileges of the Nuxeo's user:

```
jstatd -J-Djava.security.policy=jstat.permissions
```

2. On your SSH connection, configure a local “dynamic” application-level port forwarding:

```
ssh -D 6767 remote-server-host
```

3. Run `jvisualvm` on your local host and in the network options, enable the manual proxy settings and configure a socks proxy:

```
localhost:6767
```

4. Now in `jvisualvm`, add a remote host for the `remote-server-host`.
   You should get the list of Java processes ran by Nuxeo's user remotely.
5. Identify Nuxeo's Tomcat and launch a connection.

In the given `jstat` command line, we reference the Java security configuration file `jstat.permissions`. Here is its content:

```
grant codebase "file:${java.home}/../lib/tools.jar" {
    permission java.security.AllPermission;
};
```

Supervision

About Nuxeo Management Infrastructure

Nuxeo Runtime Management

The bundle `nuxeo-runtime-management` provides a management infrastructure based on Java Simon.

Two main types of objects are managed:

- Counters,
- StopWatchs.

`nuxeo-runtime-management` also provides a way to publish JMX Resources (in addition of the Counters and StopWatches that are automatically published): the factories extension point can be used to contribute new JMX resources.

Counters and StopWatches are accessible:

- via JMX,
- via the Admin Center.

Nuxeo Core Management

Nuxeo Core Management uses Nuxeo Runtime Management:

- to expose counters on Events,
- to publish a JMX Bean to manage events and Events handler,
- to publish a JMX Bean to manage "Administrative Status" and Probes.

Administrative Status are a way to define cluster-wide or instance-wide named variable that can be used to manage the status of a running platform:

- turn on/off a node of the cluster,
- display a message to all users of the platform,
- ...

Administrative Status can be configured and declared via an extension point.

Core Management also adds the concept of probes that can be used to run a test on the target deployed platform. Probes can be used to check that all part of the architecture are actually running for real:

- check LDAP access,
- check instance availability,
- check VCS access,
- ...

Probes can be defined via a dedicated extension point.

Probes and Administrative status are accessible:

- via JMX,
- via REST,
- via the Admin Center.

On this page

Existing Metrics and Counters

By default, Nuxeo Platform comes with a set of counters, StopWatches and probes.

Event System

All operations executed against the repository will trigger events. For this reason, monitoring events is a good way to have an idea about the activity of the platform.

There are by default three defined counters:

- `event.create`: counts the number of created Document/Version,
- `event.update`: counts the number of updated Documents,
- `event.remove`: counts the number of removed Documents/Version.

A `EventMonitoring` service is also exposed via JMX (and as a Java Nuxeo Service). This `EventMonitoring` service allows to:

- enable/disable all synchronous event listeners,
- enable/disable all asynchronous event listeners,
- enable/disable tracking of listeners execution,
- enable/disable bulk mode,
- retrieve statistics about event handler execution.

This `EventMonitoring` service can be used:

- for tuning the platform during mass import,
- for profiling execution of listeners,
- for monitoring.

Repository

Counters and Metrics

The repository exposed by default three counters about cache usage:

- `cache.access`,
- `cache.hits`,
- `cache.size`.

These counters are updates every 200 access to avoid having bad performance impacts.

If `org.nuxeo.vcs.cache.statistics` is set to true (via the System information tab of the Admin Center or via nuxeo.conf), there are also two StopWatch that are defined:

- `cache.gets`: time to get an entry from cache,
- `sor.gets`: time to bulk get entries from DB.

Services

`SQLStorage` service provides a JMX API to:

- gather information about the repository (active sessions, ...),
- do administration operations (Flush caches, Start Binaries GC ...).

Transactions Management

To have more information about the repository and transaction management, you can also deploy the additional bundle `nuxeo-core-management-jtajca`.

This bundle provides:

- StopWatch for transactions,
- TransactionMonitoring Service,
- JCA Connectionpool Service.

Web Layer

Two counters are published by default in JMX:

- `web.requests`: number of HTTP requests served,
- `web.sessions`: number of HTTP sessions.

Probes and Administrative Status

Nuxeo comes with a set of default probes and administrative status. Both can be seen and managed from JMX and from Admin Center.

Administrative Status

By default only three status are defined :

- `nuxeoInstance`: indicates if a Nuxeo instance (cluster node) is active of not,
- `adminMessage`: message to be displayed to all users,
- `smtpService`: defines if SMTP gateway can be used.

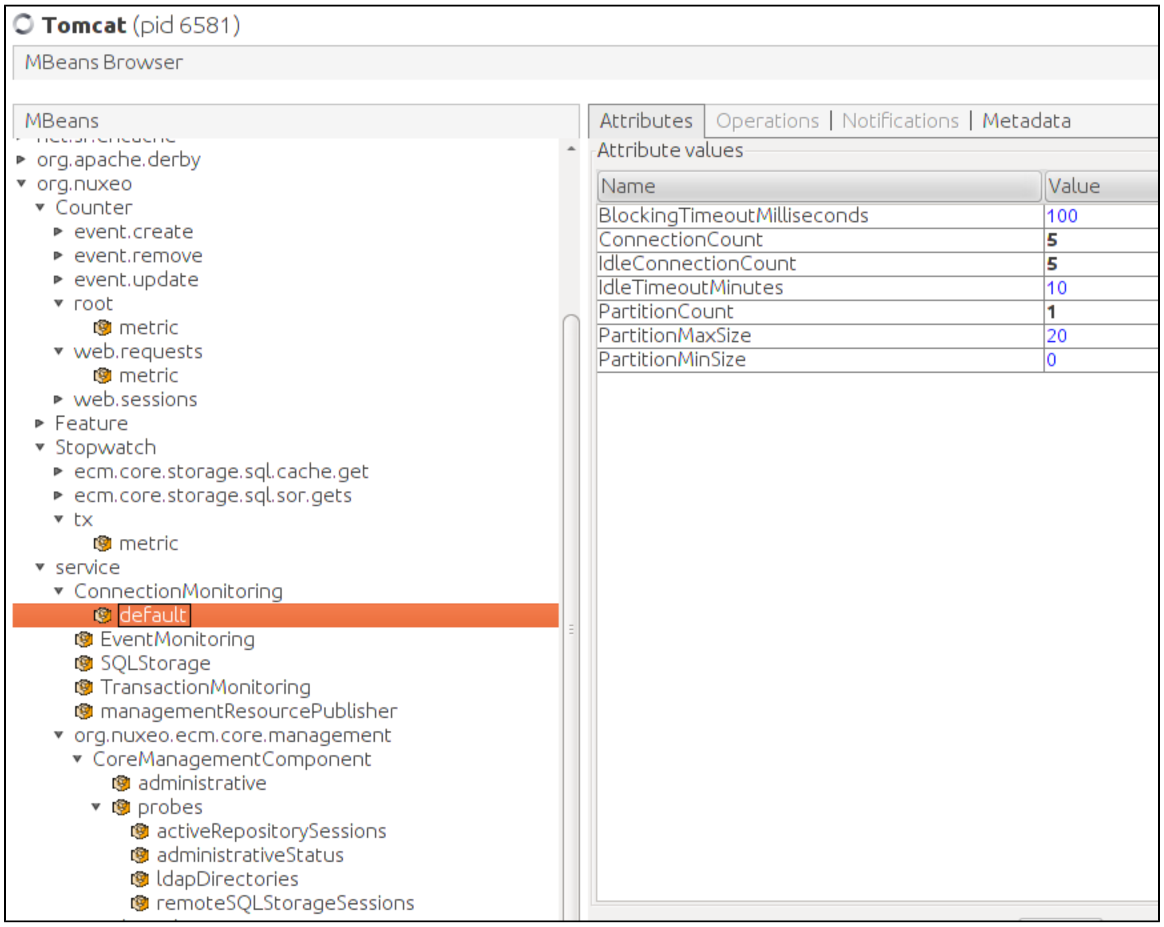
Probes

By default four probes are defined:

- `adminStatus`: checks local instance enable flag (checks `nuxeoInstance` adminsitrative status),
- `activeRepositorySession`: returns the number of active sessions per repository,
- `ldapDirectory`: check LDAP connectivity,
- `remoteSQLStorageSession`: number of remove VCS client connected (only used in VCS client/server mode that is not enabled by default).

JMX Access

You can use JVisualVM or similar tool to access Nuxeo JMX interface.



Tomcat (pid 6581)

MBeans Browser

```
MBeans
  ▸ org.apache.derby
  ▾ org.nuxeo
    ▾ Counter
      ▸ event.create
      ▸ event.remove
      ▸ event.update
      ▾ root
          metric
      ▾ web.requests
          metric
      ▸ web.sessions
    ▸ Feature
    ▾ Stopwatch
      ▸ ecm.core.storage.sql.cache.get
      ▸ ecm.core.storage.sql.sor.gets
      ▾ tx
          metric
    ▾ service
      ▾ ConnectionMonitoring
          default
        EventMonitoring
        SQLStorage
        TransactionMonitoring
        managementResourcePublisher
      ▾ org.nuxeo.ecm.core.management
        ▾ CoreManagementComponent
            administrative
          ▾ probes
              activeRepositorySessions
              administrativeStatus
              ldapDirectories
              remoteSQLStorageSessions
```

Attributes | Operations | Notifications | Metadata

Attribute values

Name	Value
BlockingTimeoutMilliseconds	100
ConnectionCount	5
IdleConnectionCount	5
IdleTimeoutMinutes	10
PartitionCount	1
PartitionMaxSize	20
PartitionMinSize	0

JMX Remote access is by default disabled. You can activate it by adding the required option (for example in nuxeo.conf)

```
-Dcom.sun.management.jmxremote
```

However, you will then have to manage security for this access, since there is no authentication by default.

Using the Admin Center

Inside the Admin Center there are two sections that are related to monitoring: Activity and Monitoring

Activity

The Activity section provides access to:

- a view that displays HTTP counters (requests and sessions),
- a view on audit logs,
- activity charts based on web and repository counters.

Monitoring

The Monitoring sections provides access to:

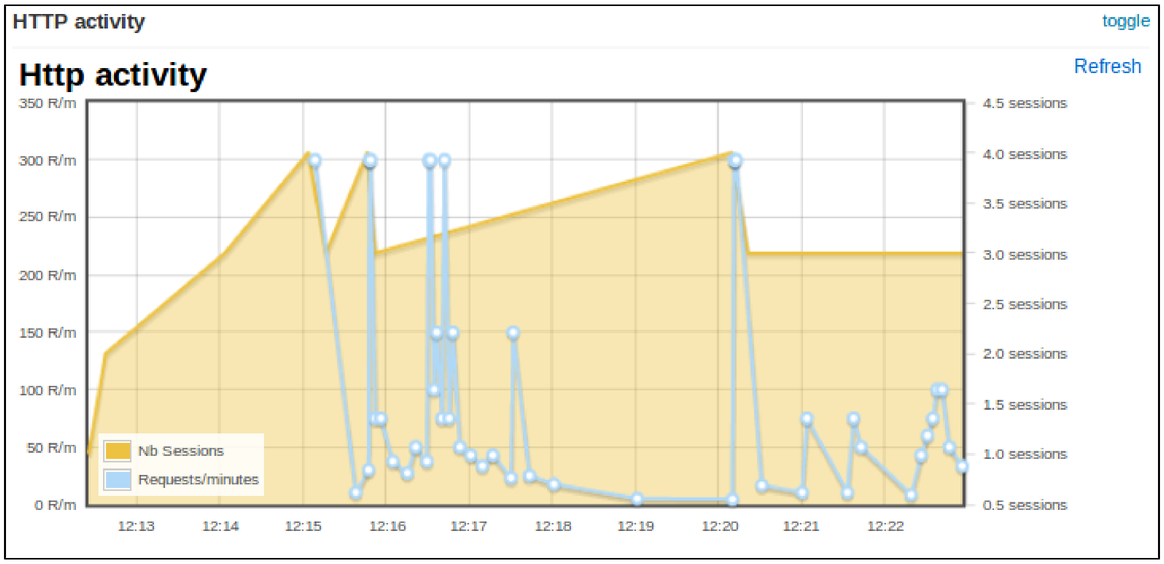
- a view on Administrative Status (view / edit),
- a view on probes (view/run),
- a view that allows to enable Event Listener statistic gathering.

Rest Access

Counters

Counter are exposed via Automation API Counters.GET

This API is used inside the Admin Center to be able to generate the small graphs with an OpenSocial gadget.



Sample CURL call:

```
curl -H 'Content-Type:application/json+nxrequest' -X POST -d
'{"params":{"counterNames":"org.nuxeo.web.requests"}}' -u
Administrator:Administrator
http://localhost:8080/nuxeo/site/automation/Counters.GET
```

Transactions and Connections

Troubleshooting issues with connections with transactional resources (databases) can be done with the optional bundle `nuxeo-core-management-jtajca`. Just download it from our Nexus and put it in `NUXEO_HOME/nxserver/plugins.` Then restart the Nuxeo server.

Logging Transactional Events

Configure log4j in `NUXEO_HOME/lib/log4j.xml` by adding the following keywords to your appender conversion pattern `%t` for the thread name and `%X` for the logging context map:

```
<param name="ConversionPattern" value="%d{ISO8601} %t %-5p [%c] %m %X%n" />
```

You should also add a new category if you want the traces being enabled:

```
<category name="org.nuxeo.ecm.core.management.jtajca">
  <priority value="TRACE" />
</category>
```

At this stage, once a transaction is started or a connection is opened, their identifiers are put in a context map for the logger. By adding the `%X` key word, you've requested to print them each a message is logged. The transactions and connections will also be logged. You should add additional log statements at level debug or trace around the code you want to monitor.

Monitoring Transactional Resources

You should enable JMX for being able to poll the mbean attributes. In `NUXEO_HOME/bin/nuxeo.conf` uncomment the JMX options.

You should note that the these settings open a **security hole** on the server and should not be left as this in production.

```
# Enable JMX
JAVA_OPTS=$JAVA_OPTS -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=1089 -Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

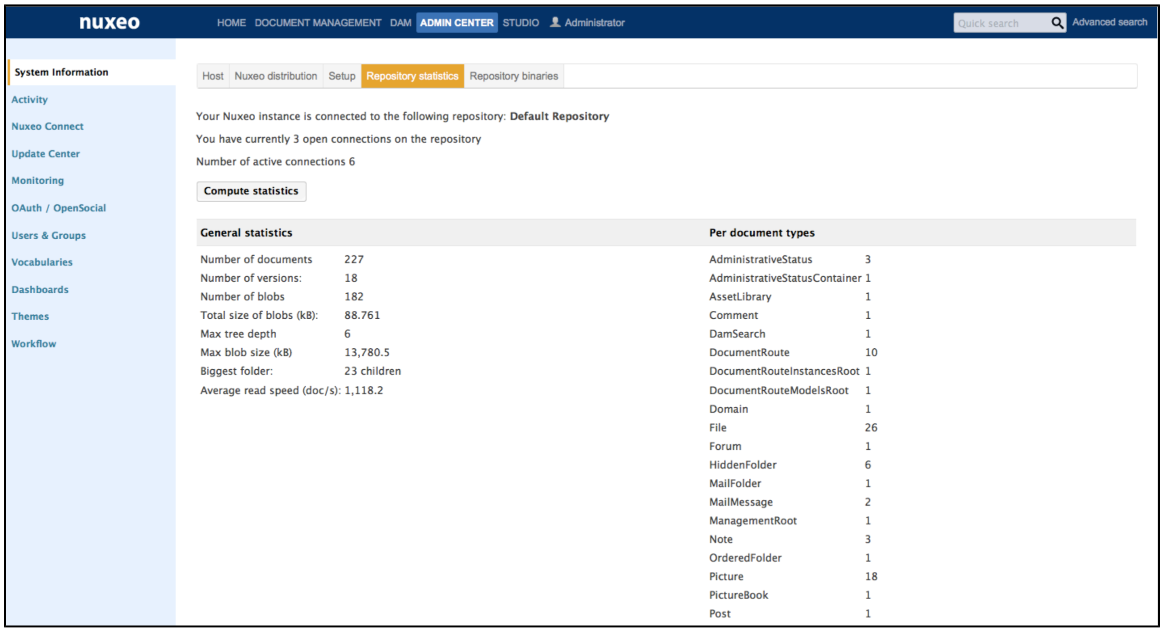
You've got two new beans in JMX that will provide you the way of monitoring the transactions and connections being used in your system. Just point your JMX browser (such as JVisualVM which is part of the JDK) to the server and looks to the following beans :

- `org.nuxeo:name=ConnectionMonitoring,type=service,repositoryName=default`
- `org.nuxeo:name=TransactionMonitoring,type=service`

The transaction monitoring provides useful information about the transactions in progress in your application. Also it gives you global counters and information about the last committed and rollbacked transaction in the system. The connection monitoring provide you a way of configuring the pool connection size used by the Nuxeo storage. It gives you also access to global counters about the connection usage.

Counting documents

If you want to know how many documents you have in your repository, you can go to the Admin Center, and run a "repostat" by clicking on **Comp ute statistics** in the **Repository Statistics** tab.



Your Nuxeo instance is connected to the following repository: **Default Repository**

You have currently 3 open connections on the repository

Number of active connections 6

General statistics	
Number of documents	227
Number of versions:	18
Number of blobs	182
Total size of blobs (kB):	88.761
Max tree depth	6
Max blob size (kB)	13,780.5
Biggest folder:	23 children
Average read speed (doc/s):	1,118.2

Per document types	
AdministrativeStatus	3
AdministrativeStatusContainer	1
AssetLibrary	1
Comment	1
DamSearch	1
DocumentRoute	10
DocumentRouteInstancesRoot	1
DocumentRouteModelsRoot	1
Domain	1
File	26
Forum	1
HiddenFolder	6
MailFolder	1
MailMessage	2
ManagementRoot	1
Note	3
OrderedFolder	1
Picture	18
PictureBook	1
Post	1

You can also install the Nuxeo Quota plugin, see the user documentation about it.

Nuxeo Shell

Availability Info
The Nuxeo Shell can work against any Nuxeo distributions since the **5.4.0.1** version of the Nuxeo Platform and deprecates the old RMI-based Shell.

Nuxeo Shell works on top of Nuxeo Content Automation Client and is using the REST API provided by the Automation framework. Because it is using HTTP to communicate with the server the Shell can be used against any Nuxeo distribution which includes the automation framework.

This also means, most of the Shell commands are implemented as remote Automation operations.

Downloads
The Nuxeo Shell is available at the Admin Center, in the "Monitor" tab and also as a WebEngine site at http://host:port/nuxeo/site/shell.

Nuxeo Shell can be installed in Eclipse adding the following repository: Nuxeo ECR - http://osgi.nuxeo.org/p2/ecr/ide/, then install "ECR Shell Feature".

Download the last released version of Nuxeo Shell for a manual install. You can browse available versions at https://maven.nuxeo.org (or pick latest).

Overview

Nuxeo Shell comes with a lot of features to improve your experience using the command line. Also, it provides high pluggability so you can extend it and add new commands in a simple way.

Here is a list of the most important features.

In this section

Interactive Execution

The Nuxeo Shell uses `jline` library for interactive mode. When you launch the Shell you will automatically enter the interactive mode. This mode provides a prompt and auto completion so you can easily browse a Nuxeo repository and execute commands.

Batch Execution

You can also launch the Shell in batch mode. In this mode you can specify a list of commands to be executed and then to return the control back to the terminal.

There are 3 batch modes available:

1. Run commands from a file
2. Run commands from standard input.
3. Run commands specified on the command line - this is a convenient way to run a short list of commands.

See the Shell Batch Mode page for more details.

Namespaces

Namespaces are commands registry that you can register in the Shell. A command registry is useful to bring inside the Shell a new set of commands for a specific context without having name clash problems with already existing commands.

To switch to a different namespace use the `use` command.

So for example, you may want to have `ls` command that is listing the content of a Nuxeo Folder when connected to a remote server but still use the `ls` command when switching to the **local** context to be able to list the content of a directory on the local file system.

Also, namespaces are hierarchical so a namespace my extend another one to adds more commands. Available namespaces are setup by the features installed in the Shell. By default, Nuxeo Shell provides the following default namespaces:

1. global - the global namespace. This provides global commands like `help`, 'use', 'cmds' etc.
2. local - provides file system commands like: `ls`, `cd`, `cat`, etc. Extends the global namespace.
3. remote - provides remote commands against a Nuxeo server. Extends the global namespace. Available only after connecting to a remote server.
4. automation - expose remote automation operations as commands. Available only after connecting to a remote server.

Auto Completion

Auto completion support is provided by **jline**. The Shell is leveraging this so that you can auto complete:

1. command names.
2. parameter names.
3. parameter values (when the command provide completion).

There are several type of objects that supports completion (like, file, document, etc.) but you can add more by implementing new completors.

Scripting

Security Warning
Because of potential security issues the scripting feature is available only when logged in as Administrator

Nuxeo Shell is providing scripting capabilities through Groovy or Mvel. You can execute thus your Groovy scripts on the server JVM. There are two ways of executing scripts:

1. Either invoke a script file from your file system using the **script** command.
2. Either write a command that execute a script (instead of executing a remote automation operation).

Documentation Generation

Command documentation pages are automatically generated by the Shell by introspecting the command annotations.

Also, if you want to add more details like in-depth explanation of the command, examples etc. then you can write this in a text file and the Shell will automatically append it to the generated documentation.

When writing custom documentation you can use tags like {header}, {bold}, {red}, {green} etc. to control the formatting on the terminal.

Automation Chain Execution

You can execute any automation chain exposed by the server by using the `run` or `runonfile` commands. This is useful since you can create your administration commands in Nuxeo Studio by creating Automation Chains.

Direct Operation Execution

The automation mode provides Shell commands for any operation available on the server. These commands are automatically generated from operation descriptors.

This mode should be used only to debug operations or to execute some operations not exposed through specialized commands by the Shell.

Experimental
Because of the Shell environment not every automation operation command will correctly work. You should use regular operations instead.

Exception Handling

In the interactive mode unchecked exceptions are printed in red on the screen. Checked exceptions are not printed - only the exception message is printed out. If you need to see the stack trace of the last exception type **trace**.

Extensibility

A Shell feature is providing Shell configuration such as namespaces, commands, completors, etc. To extend the capabilities of the Shell you can register a new feature which will install the new capabilities into the Shell at boot time.

The Shell can be extended wither by modifying the Nuxeo Shell JAR either by adding on the classpath JARs containing additional Nuxeo Shell features.

See Extending the Shell section for more details.

Usage

Launching the Shell

You can launch the Shell either by running the command:

```
java -jar nuxeo-shell.jar
```

either by running:

```
java -cp nuxeo-shell.jar org.nuxeo.shell.Main
```

The difference is that the first command will launch the Shell in a Swing based terminal (i.e. a desktop application) while the second one will launch the Shell inside your current terminal.
On some Operating Systems like Windows double clicking the nuxeo-shell.jar will launch the Shell in a Swing based terminal.

When launching the Shell the **local** namespace is automatically activated.

When connecting to a remote server the Shell will automatically switch to **remote** namespace. When disconnecting it will go back to the **local** namespace.

To switch between namespaces just use the **use** command. For example if you are in the **remote** namespace you can switch to the **local** on e by executing the command:

```
use local
```

Getting the Shell Version

You can launch the Shell using

```
java -jar nuxeo-shell.jar --version
```

to get information about the Shell version and the minimal server version required by the Shell to correctly run against a remote server.

You can also have this information by using the **version** command.

Connecting to a Server

To connect to a remote Nuxeo Server you must specify the Automation Service URL.

This URL is in the form:

```
http://host:port/nuxeo/site/automation
```

For example, in the case of a local server your URL will be:

```
http://localhost:8080/nuxeo/site/automation
```

To connect to a server you should use the global **connect** command. This command require 3 parameters:

1. the URL of the remote automation service.
2. the username to login
3. the password to login

You can either pass these arguments to the connect command itself or through the command line when starting the Shell.

You can use `-u` for the username, `-p` for the password and the URL should be given as argument.

If password is not specified you will be prompted for a password when in interactive mode.

Example

Here is an example of a short session:

After launching the Shell you are in **local** mode. So you can use the provided file system commands like:

- `ls` - to list the content of the current directory.
- `cd`, `pushd`, `popd` - to change the current directory.
- `cat`, `mv`, `cp`, etc. for other file based operations.

To connect to a remote server type:

```
connect http://localhost:8080/nuxeo/site/automation -u Administrator
```

After the connection is done you are automatically switched in **remote** namespace.

So doing now a `ls` will list the content of the current document. (which for now the root document).

To switch back in **local** namespace type:

```
use local
```

To show the current namespace name type:

```
use
```

Note
When doing file based auto-completion this will be relative to the current directory (that you can change using `cd`, `pushd`, `popd` wh en in local namespace). The same for document based auto-completion.

Nuxeo Shell Command Index

Here is a list of namespaces available in the Nuxeo Shell, each namespace providing an index of its commands.

The command index is generated using the shell command

```
help -export [file] -ns [namespace]
```

Built-in Commands

Namespace: *global*
Basic commands provided by the shell

Index

commands

NAME
commands – Print a list of available commands

SYNTAX

```
commands
```

ALIASES
cmds

exit

NAME
exit – Exit the interactive shell

SYNTAX

```
exit [code]
```

ALIASES
quit

ARGUMENTS

- code - [optional] -The exit code. Must be a positive number otherwise 0 is assumed. Defaults to 0.

help

NAME
help – The help command

SYNTAX

```
help [options] [command]
```

OPTIONS

- -export - If used export all the commands available in a wiki format to the given file. If a directory is given the export will be made in file help.wiki in that directory.
- -ns - [optional] - to be used to filter commands by namespaces when generating the documentation. By default all namespaces are dumped.

ARGUMENTS

- command - [optional] -the name of the command to get help for

install

NAME
install – Install a SH script to launch the shell in the terminal. Available only for UNIX systems.

SYNTAX

```
install [file]
```

ARGUMENTS

- file - [optional] -the file where to install the shell script. If not used the script will be printed on stdout.

interactive

NAME
interactive – Interactive shell

SYNTAX

```
interactive
```

settings

NAME
settings – Print or modify the shell settings.

SYNTAX

```
settings [options] [name] [value]
```

OPTIONS

- -reset - [flag] - Reset settings to their defaults. Need to restart shell.

ARGUMENTS

- name - [optional] -The variable to print or set.
- value - [optional] -The variable value to set.

trace

NAME
trace – Print the last error stack trace if any

SYNTAX

```
trace
```

use

NAME
use – Switch the current command namespace. If no namespace is specified the current namepsace name is printed.

SYNTAX

```
use [name]
```

ARGUMENTS

- name - [optional] -The command namespace to use

version

NAME
version – Print Nuxeo Shell Version

SYNTAX

```
version
```

Related pages

- Nuxeo Shell (Nuxeo Installation and Administration - 5.8)
- Built-in Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Automation Commands (Nuxeo Installation and Administration - 5.8)
- Filesystem Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)
- Shell Commands (Nuxeo Platform Developer Documentation - 5.8)
- Shell Features (Nuxeo Platform Developer Documentation - 5.8)
- Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)
- Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)
- Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)
- Configuration Commands (Nuxeo Installation and Administration - 5.8)

• Filesystem Commands

Namespace: *local*
Commands available on the local file system

Index:

- cat
- cd
- cp
- ls
- mkdir
- mv
- popd
- pushd
- pwd
- rm
- touch

cat

NAME
cat – Print the content of a file

SYNTAX

```
cat [file]
```

ARGUMENTS

- file - [optional] -The file to print

cd

NAME
cd – Change the local working directory

SYNTAX

```
cd file
```

ARGUMENTS

- file - [required] - A local directory to change to

cp

NAME
cp – Copy a file or directory

SYNTAX

```
cp [options] source destination
```

OPTIONS

- -r - [flag] - Recursive copy directories

ARGUMENTS

- source - [required] - The file to copy
- destination - [required] - The target file

ls

NAME
ls – List file names in a local directory

SYNTAX

```
ls [file]
```

ARGUMENTS

- file - [optional] -A local directory to list its content. Defaults to the working directory

mkdir

NAME
mkdir – Create a directory in local file system

SYNTAX

```
mkdir file
```

ARGUMENTS

- file - [required] - The directory path to create

mv

NAME
mv – Rename a file or directory

SYNTAX

```
mv source destination
```

ARGUMENTS

- source - [required] - The file to rename
- destination - [required] - The target file

popd

NAME
popd – Pop working directory

SYNTAX

```
popd
```

pushd

NAME
pushd – Push a new local working directory

SYNTAX

```
pushd file
```

ARGUMENTS

- file - [required] - A local directory to push

pwd

NAME
pwd – Print the local working directory

SYNTAX

```
pwd [options]
```

OPTIONS

- -s - [flag] - Use this flag to show the working directory stack

rm

NAME
rm – Remove a file or directory

SYNTAX

```
rm [options] file
```

OPTIONS

- -r - [flag] - Recursive remove directories

ARGUMENTS

- file - [required] - The directory or file to delete

touch

NAME
touch – Touch a file

SYNTAX

```
touch file
```

ARGUMENTS

- file - [required] - The file to touch

Related pages

- Nuxeo Shell (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Command Index (Nuxeo Installation and Administration - 5.8)
- Built-in Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Automation Commands (Nuxeo Installation and Administration - 5.8)
- Filesystem Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)
- Shell Commands (Nuxeo Platform Developer Documentation - 5.8)
- Shell Features (Nuxeo Platform Developer Documentation - 5.8)
- Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)
- Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)
- Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)
- Configuration Commands (Nuxeo Installation and Administration - 5.8)

Nuxeo Server Commands

Namespace: *remote*
High level commands exposed by a remote Nuxeo Server

Index:

- tree
- unlock
- update

audit

NAME
audit – Run a query against audit service

SYNTAX

```
audit [options] query
```

OPTIONS

- -s - Use this to change the separator used in query variables. The default is ','
- -ctx - Use this to set query variables. Syntax is: "k1=v1,k1=v2"
- -max - The max number of rows to return.
- -page - The current page to return. To be used in conjunction with -max.
- -f - Use this to save results in a file. Otherwise results are printed on the screen.

ARGUMENTS

- query - [required] - The query to run. Query is in JPQL format

USAGE

The -page parameter can be used in conjunction with -max parameter to paginate the query result.
The specify the first page use 1 as value, for the second page use 2 and so on.

When saving results in a file - they are in JSON format - and dates are specified using a long value timestamp.
Results printed on the screen are printed in tab separated columns:
eventId category eventDate principal docUUID docType docLifeCycle comment

EXAMPLES

Using date literals in your query:

```
audit "FROM LogEntry log WHERE log.eventDate > timestamp('2010-11-10 00:00:00')"
```

Using pagination:

```
audit "FROM LogEntry log WHERE log.category='studio' ORDER BY log.eventDate DESC" -max
20 -page 1
```

Using query variables:

```
audit "FROM LogEntry log WHERE log.category='studio' AND log.eventDate > :startDate"
-ctx "startDate={d 2010-11-10}"
```

or

```
audit "FROM LogEntry log WHERE log.category='studio' AND log.eventDate > :startDate"
-ctx "startDate={d 2010-11-10 00:00:00}"
```

Note that query variable keys must be prefixed with "audit.query." to avoid name clash with other keys in the context.

cat

NAME

cat – Print document details

SYNTAX

```
cat [options] [doc]
```

OPTIONS

- -all - [flag] - Include all schemas. The -schemas attribute will be ignored if used in conjunction with this one.
- -schemas - A filter of schemas to include in the document. Use * for all schemas.

ARGUMENTS

- doc - [optional] -The document to print. To use UIDs as references you should prefix them with 'doc:'

cd

NAME
cd – Change the context document

SYNTAX

```
cd doc
```

ARGUMENTS

- doc - [required] - A reference to the new context document to use. To use UID references prefix them with 'doc:'.

connect

NAME
connect – Connect to a remote automation server

SYNTAX

```
connect [options] [url]
```

OPTIONS

- -p - The password
- -u - The username

ARGUMENTS

- url - [optional] -The URL of the automation server

cp

NAME
cp – Copy a document

SYNTAX

```
cp [options] src dst
```

OPTIONS

- -name - A new name for the copy. I not specified preserve the source name

ARGUMENTS

- src - [required] - The document to copy. To use UID references prefix them with 'doc:'.
- dst - [required] - The target parent. To use UID references prefix them with 'doc:'.

disconnect

NAME
disconnect – Close current connection to server. If not connected nothing is done.

SYNTAX

```
disconnect
```

fire

NAME
fire – Fire a core event in the context of the given document

SYNTAX

```
fire event [doc]
```

ARGUMENTS

- event - [required] - The event to fire. This parameter is required.
- doc - [optional] -The context document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

getfile

NAME
getfile – Get a document attached file

SYNTAX

```
getfile [options] [doc]
```

OPTIONS

- -todir - An optional target directory to save the file. The default is the current working directory.
- -xpath - The XPath of the blob property to get. Defaults to the one used by the File document type.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

getfiles

NAME
getfiles – Get all the files attached to a document

SYNTAX

```
getfiles [options] [doc]
```

OPTIONS

- -todir - An optional target directory to save the file. The default is the current working directory.
- -xpath - The XPath of the blob property to get. Defaults to the one used by the File document type.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document content is used. To use UID references prefix them with 'doc:'.

getp

NAME
getp – Get the value of a document property

SYNTAX

```
getp [options] [doc]
```

OPTIONS

- -xpath - The XPath of the property to get. This parameter is required.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

getrel

NAME
getrel – Get relations between two documents

SYNTAX

```
getrel [options] [doc]
```

OPTIONS

- -in - [flag] - Is the document the relation object?
- -predicate - The relation predicate - requested.
- -out - [flag] - Is the document the relation subject? This flag is by default on true.

ARGUMENTS

- doc - [optional] -The document involved in the relation

lcstate

NAME
lcstate – Set or view the current life cycle state of a document

SYNTAX

```
lcstate [options] [doc]
```

OPTIONS

- -set - If specified The new life cycle state. If not specified then in write mode the local ACL is used and in view mode all ACLs are printed.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

lock

NAME
lock – Lock a document

SYNTAX

```
lock [options] [doc]
```

OPTIONS

- -key - An optional lock key. If not specified the default one is used.

ARGUMENTS

- doc - [optional] -The document to lock. If not specified the current document is used. To use UID references prefix them with 'doc:'.

ls

NAME
ls – List children documents

SYNTAX

```
ls [options] [doc]
```

OPTIONS

- -uid - [flag] - If used the documents will be printed using the document UID.

ARGUMENTS

- doc - [optional] -A document to list its content. If not specified list the current document content. To use UID references prefix them with 'doc:'.

mkdir

NAME
mkdir – Create a document of the given type

SYNTAX

```
mkdir [options] type path
```

OPTIONS

- -title - An optional document title.

ARGUMENTS

- type - [required] - The document type
- path - [required] - The document path

mkrel

NAME
mkrel – Create a relation between two documents

SYNTAX

```
mkrel [options] subject object
```

OPTIONS

- -predicate - The relation predicate - requested.

ARGUMENTS

- subject - [required] - The subject of the relation
- object - [required] - The object of the relation

mv

NAME
mv – Move a document

SYNTAX

```
mv [options] src dst
```

OPTIONS

- -name - A new name for the document. I not specified preserve the source name

ARGUMENTS

- src - [required] - The document to move. To use UID references prefix them with 'doc:'.
- dst - [required] - The target parent. To use UID references prefix them with 'doc:'.

perms

NAME
perms – Set or view permissions on a document

SYNTAX

```
perms [options] [doc]
```

OPTIONS

- -grant - If used the ACL will be modified by granting the specified permission on the specified user. The grant value format is "user:permission".
- -deny - If used the ACL will be modified by denying the specified permission on the specified user. The deny value format is "user:permission".
- -remove - [flag] - Remove the given ACL.
- -acl - The ACL to view or modify. If not specified then in write mode the local ACL is used and in view mode all ACLs are printed.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

popd

NAME
popd – Change the context document and pop the document from the navigation stack.

SYNTAX

```
popd
```

publish

NAME
publish – Publish a document into a section

SYNTAX

```
publish [options] src section
```

OPTIONS

- -override - If set to false will not override an existing published document with same name. The default is "true".

ARGUMENTS

- src - [required] - The document to copy. To use UID references prefix them with 'doc:'.
- section - [required] - The target parent. To use UID references prefix them with 'doc:'.

pushd

NAME
pushd – Change the context document and push the document on the navigation stack.

SYNTAX

```
pushd doc
```

ARGUMENTS

- doc - [required] - A reference to the new context document to use. To use UID references prefix them with 'doc:'.

putfile

NAME
putfile – Attach a file to a document

SYNTAX

```
putfile [options] file [doc]
```

OPTIONS

- -xpath - The XPath of the blob property to set. Defaults to the one used by the File document type.

ARGUMENTS

- file - [required] - The file to upload
- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

pwd

NAME
pwd – Print the current context document

SYNTAX

```
pwd [options]
```

OPTIONS

- -s - [flag] - Use this flag to show the context documents stack

query

NAME
query – Query documents

SYNTAX

```
query [options] [query]
```

OPTIONS

- -uid - [flag] - If used the matching documents will be printed using the document UID.

ARGUMENTS

- query - [optional] -The document path

EXAMPLES

```
query "SELECT * FROM Document WHERE ecm:primaryType='Folder'"
```

```
query -uid "SELECT * FROM Document WHERE ecm:primaryType=\"Folder\""
```

rename

NAME
rename – Rename a document

SYNTAX

```
rename [options] [doc]
```

OPTIONS

- -name - A new name for the document. This parameter is required.

ARGUMENTS

- doc - [optional] -The document to rename. To use UID references prefix them with 'doc:'.

rm

NAME
rm – Remove a document

SYNTAX

```
rm [path]
```

ARGUMENTS

- path - [optional] -The document path. To use UID references prefix them with 'doc:'.

rmfile

NAME
rmfile – Remove an attached file from a document

SYNTAX

```
rmfile [options] [doc]
```

OPTIONS

- -xpath - The XPath of the blob property to remove. Defaults to the one used by the File document type.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

run

NAME
run – Run a server automation chain that accepts a document or void input

SYNTAX

```
run [options] chain [doc]
```

OPTIONS

- -s - Use this to change the separator used in context variables. The default is ','
- -ctx - Use this to set execution context variables. Syntax is: k1=v1,k1=v2
- -void - [flag] - Use this to avoid having the server sending back the result.

ARGUMENTS

- chain - [required] - The chain to run
- doc - [optional] -A reference to the new context document to use. To use UID references prefix them with 'doc:'.

runonfile

NAME
runonfile – Run a server automation chain that accepts a file as an input

SYNTAX

```
runonfile [options] chain file
```

OPTIONS

- -s - Use this to change the separator used in context variables. The default is ','
- -ctx - Use this to set execution context variables. Syntax is: k1=v1,k1=v2
- -void - [flag] - Use this to avoid having the server sending back the result.

ARGUMENTS

- chain - [required] - The chain to run
- file - [required] - A reference to the new context document to use. To use UID references prefix them with 'doc:'.

script

NAME
script – Run a script on the server

SYNTAX

```
script [options] file
```

OPTIONS

- -s - Use this to change the separator used in context variables. The default is ','
- -ctx - Use this to set execution context variables. Syntax is: "k1=v1,k1=v2"

ARGUMENTS

- file - [required] - The script file. Must have a .mvel or .groovy extension

setp

NAME
setp – Set a property on a document

SYNTAX

```
setp [options] [doc]
```

OPTIONS

- -value - The property value. If not specified the current property value is removed.

- -xpath - The XPath of the property to set. This parameter is required.

ARGUMENTS

- doc - [optional] -The target document. If not specified the current document is used. To use UID references prefix them with 'doc:'.

tree

NAME
tree – List a subtree

SYNTAX

```
tree [doc]
```

ARGUMENTS

- doc - [optional] -A document to list its subtree. If not specified list the current document subtree. To use UID references prefix them with 'doc:'.

unlock

NAME
unlock – Unlock a document

SYNTAX

```
unlock [doc]
```

ARGUMENTS

- doc - [optional] -The document to unlock. If not specified the current document is used. To use UID references prefix them with 'doc:'.

update

NAME
update – Update document properties

SYNTAX

```
update [options] [properties] [path]
```

OPTIONS

- -s - Use this to change the separator used in properties. The default is ','

ARGUMENTS

- properties - [optional] -The properties to update.
- path - [optional] -The document path

Related pages

Nuxeo Shell (Nuxeo Installation and Administration - 5.8)

Filesystem Commands (Nuxeo Installation and Administration - 5.8)

- ## Nuxeo Automation Commands

Namespace: *automation* Commands exposed by the Nuxeo Server through automation

Audit.Log

- Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)
- Shell Commands (Nuxeo Platform Developer Documentation - 5.8)
- Shell Features (Nuxeo Platform Developer Documentation - 5.8)
- Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)
- Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)
- Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)
- Configuration Commands (Nuxeo Installation and Administration - 5.8)

NAME
Audit.Log – Log events into audit for each of the input document. The operation accept as input one ore more documents that are returned back as the output.

SYNTAX

```
Audit.Log [options] [the input document(s)]
```

OPTIONS

- -event -
- -ctx - Can be used to inject context properties in Java properties format
- -category -
- -void - [flag] - If void the server will not return the result back
- -comment -

ARGUMENTS

- the input document(s) - [optional] -null

Audit.Query

NAME
Audit.Query – Execute a JPA query against the Audit Service. This is returning a blob with the query result. The result is a serialized JSON array. You can use the context to set query variables but you must prefix using 'audit.query.' the context variable keys that match the ones in the query.

SYNTAX

```
Audit.Query [options]
```

OPTIONS

- -maxResults -
- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -pageNo -
- -query - Be sure to use JPA Query. NXSQL will not work with audit log files.

Auth.LoginAs

NAME
Auth.LoginAs – Login As the given user. If no user is given a system login is performed. This is a void operations - the input will be returned back as the output.

SYNTAX

```
Auth.LoginAs [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Auth.Logout

NAME

Auth.Logout – Perform a logout. This should be used only after using the Login As operation to restore original login. This is a void operations - the input will be returned back as the output.

SYNTAX

```
Auth.Logout [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Blob.Attach

NAME
Blob.Attach – Attach the input file to the document given as a parameter. If the XPath points to a blob list then the blob is appended to the list, otherwise the XPath should point to a blob property. If the save parameter is set the document modification will be automatically saved. Return the blob.

SYNTAX

```
Blob.Attach [options] the input file(s)
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -document -
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input file(s) - [required] - null

Blob.Create

NAME
Blob.Create – Creates a file from a given URL. The file parameter specifies how to retrieve the file content. It should be an URL to the file you want to use as the source. You can also use an expression to get an URL from the context. Returns the created file.

SYNTAX

```
Blob.Create [options]
```

OPTIONS

- -mime-type -
- -file -
- -ctx - Can be used to inject context properties in Java properties format
- -filename -
- -void - [flag] - If void the server will not return the result back
- -encoding -

In this section

- Blob.Pop
- Blob.PopList
- Blob.Post
- Blob.Pull
- Blob.PullList
- Blob.Push
- Blob.PushList
- Blob.Remove
- Blob.Set
- Blob.SetFilename
- Blob.ToFile
- Blob.ToPDF
- Context.FetchDocument
- Context.RestoreBlobInput
- Context.RestoreBlobsInput
- Context.RestoreDocumentInput
- Context.RestoreDocumentsInput
- Context.RunDocumentOperation
- Context.RunInputScript
- Context.RunOperation
- Context.RunScript
- Context.SetInputAsVar
- Context.SetVar
- Document.CheckIn
- Document.CheckOut
- Document.Copy
- Document.Create
- Document.CreateVersion
- Document.Delete
- Document.Fetch
- Document.FetchByProperty
- Document.Filter
- Document.GetChild
- Document.GetChildren
- Document.GetParent
- Document.GetPrincipalEmails
- Document.GetUsersAndGroups
- Document.Lock
- Document.Move
- Document.MultiPublish
- Document.Pop
- Document.PopList
- Document.Publish
- Document.Pull
- Document.PullList
- Document.Push
- Document.PushList
- Document.Query
- Document.Reload
- Document.RemoveACL
- Document.RemoveProperty
- Document.Save
- Document.SaveSession
- Document.SetACE
- Document.SetLifeCycle
- Document.SetProperty
- Document.Unlock
- Document.Update
- Notification.SendEvent
- Notification.SendMail
- Relations.CreateRelation
- Relations.GetRelations
- Workflow.CreateTask
- Workflow.GetTask
- print

Blob.CreateZip

NAME

Blob.CreateZip – Creates a zip file from the input file(s). If no file name is given, the first file name in the input will be used. Returns the zip file.

SYNTAX

```
Blob.CreateZip [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -filename -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input file(s) - [required] - null

Blob.Get

NAME
Blob.Get – Gets a file attached to the input document. The file location is specified using an XPath to the blob property of the document. Returns the file.

SYNTAX

```
Blob.Get [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Blob.GetList

NAME
Blob.GetList – Gets a list of files that are attached on the input document. The files location should be specified using the blob list property XPath. Returns a list of files.

SYNTAX

```
Blob.GetList [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Blob.Pop

NAME
Blob.Pop – Restore the last saved input file in the context input stack. This operation must be used only if a PUSH operation was previously made. Return the last *pushed* file.

SYNTAX

```
Blob.Pop [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Blob.PopList

NAME
Blob.PopList – Restore the last saved input file list in the context input stack

SYNTAX

```
Blob.PopList [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Blob.Post

NAME
Blob.Post – Post the input file to a target HTTP URL. Returns back the input file.

SYNTAX

```
Blob.Post [options] the input file(s)
```

OPTIONS

- -url -
- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input file(s) - [required] - null

Blob.Pull

NAME
Blob.Pull – Restore the last saved input file in the context input stack. This operation must be used only if a PUSH operation was previously made. Return the first *pushed* file.

SYNTAX

```
Blob.Pull [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Blob.PullList

NAME
Blob.PullList – Restore the first saved input file list in the context input stack

SYNTAX

```
Blob.PullList [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Blob.Push

NAME
Blob.Push – Push the input file on the context stack. The file can be restored later as the input using the corresponding pop operation. Returns the input file.

SYNTAX

```
Blob.Push [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input file(s) - [required] - null

Blob.PushList

NAME
Blob.PushList – Push the input file list on the context stack. The file list can be restored later as the input using the corresponding pop operation. Returns the input file list.

SYNTAX

```
Blob.PushList [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input file(s) - [required] - null

Blob.Remove

NAME
Blob.Remove – Remove the file attached to the input document as specified by the 'xpath' parameter. If the 'xpath' point to a blob list then the list will be cleared. If the file to remove is part of a list it will be removed from the list otherwise the 'xpath' should point to a blob property that will be removed. If the save parameter is set the document modification will be automatically saved. Return the document.

SYNTAX

```
Blob.Remove [options] [the input document(s)]
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Blob.Set

NAME
Blob.Set – Set the input file to the given property on the input document. If the XPath points to a blob list then the blob is appended to the list, otherwise the XPath should point to a blob property. If the save parameter is set the document modification will be automatically saved. Return the document.

SYNTAX

```
Blob.Set [options] [the input document(s)]
```

OPTIONS

- -save -
- -file -
- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Blob.SetFilename

NAME
Blob.SetFilename – Modify the filename of a file stored in the input document. The file is found in the input document given its xpath specified through the 'xpath' parameter. Return back the input document.

SYNTAX

```
Blob.SetFilename [options] [the input document(s)]
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Blob.ToFile

NAME
Blob.ToFile – Save the input blob(s) as a file(s) into the given target directory. The blob(s) filename is used as the file name. You can specify an optional **prefix** string to prepend to the file name. Return back the blob(s).

SYNTAX

```
Blob.ToFile [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -prefix -
- -void - [flag] - If void the server will not return the result back
- -directory -

ARGUMENTS

- the input file(s) - [required] - null

Blob.ToPDF

NAME
Blob.ToPDF – Convert the input file to a PDF and return the new file.

SYNTAX

```
Blob.ToPDF [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input file(s) - [required] - null

Context.FetchDocument

NAME
Context.FetchDocument – Fetch the input of the context as a document. The document will become the input for the next operation.

SYNTAX

```
Context.FetchDocument [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Context.RestoreBlobInput

NAME
Context.RestoreBlobInput – Restore the file input from a context variable given its name. Return the file.

SYNTAX

```
Context.RestoreBlobInput [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Context.RestoreBlobsInput

NAME
Context.RestoreBlobsInput – Restore the file list input from a context variable given its name. Return the files.

SYNTAX

```
Context.RestoreBlobsInput [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format

- -name -
- -void - [flag] - If void the server will not return the result back

Context.RestoreDocumentInput

NAME
Context.RestoreDocumentInput – Restore the document input from a context variable given its name. Return the document.

SYNTAX

```
Context.RestoreDocumentInput [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Context.RestoreDocumentsInput

NAME
Context.RestoreDocumentsInput – Restore the document list input from a context variable given its name. Return the document list.

SYNTAX

```
Context.RestoreDocumentsInput [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Context.RunDocumentOperation

NAME
Context.RunDocumentOperation – Run an operation chain which is returning a document in the current context. The input for the chain ro run is the current input of the operation. Return the output of the chain as a document.

SYNTAX

```
Context.RunDocumentOperation [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -id -

ARGUMENTS

- the input document(s) - [optional] -null

Context.RunInputScript

NAME
Context.RunInputScript – Run a script from the input blob. A blob containing script result is returned.

SYNTAX

```
Context.RunInputScript [options] the input file(s)
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -type -

ARGUMENTS

- the input file(s) - [required] - null

Context.RunOperation

NAME
Context.RunOperation – Run an operation chain in the current context

SYNTAX

```
Context.RunOperation [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -id -

Context.RunScript

NAME
Context.RunScript – Run a script which content is specified as text in the 'script' parameter

SYNTAX

```
Context.RunScript [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -script -
- -void - [flag] - If void the server will not return the result back

Context.SetInputAsVar

NAME
Context.SetInputAsVar – Set a context variable that points to the current input object. You must give a name for the variable. This operation works on any input type and return back the input as the output.

SYNTAX

```
Context.SetInputAsVar [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Context.SetVar

NAME
Context.SetVar – Set a context variable given a name and the value. To compute the value at runtime from the current context you should use an EL expression as the value. This operation works on any input type and return back the input as the output.

SYNTAX

```
Context.SetVar [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -value -
- -name -
- -void - [flag] - If void the server will not return the result back

Document.CheckIn

NAME
Document.CheckIn – Checks in the input document. Returns back the document.

SYNTAX

```
Document.CheckIn [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -version -
- -versionVarName -
- -void - [flag] - If void the server will not return the result back
- -comment -

ARGUMENTS

- the input document(s) - [optional] -null

Document.CheckOut

NAME
Document.CheckOut – Checks out the input document. Returns back the document.

SYNTAX

```
Document.CheckOut [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Copy

NAME
Document.Copy – Copy the input document into the given folder. The name parameter will be used as the copy name otherwise if not specified the original name will be preserved. The target folder can be specified as an absolute or relative path (relative to the input document) as an UID or by using an EL expression. Return the newly created document (the copy).

SYNTAX

```
Document.Copy [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format

- -target -
- -name -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Create

NAME
Document.Create – Create a new document in the input folder. You can initialize the document properties using the 'properties' parameter. The properties are specified as *key=value* pairs separated by a new line. The key used for a property is the property XPath. To specify multi-line values you can use a \ character followed by a new line.
Example:
`dc:title=The Document Title`
`dc:description=foo bar`
Returns the created document.

SYNTAX

```
Document.Create [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -properties -
- -name -
- -void - [flag] - If void the server will not return the result back
- -type -

ARGUMENTS

- the input document(s) - [optional] -null

Document.CreateVersion

NAME
Document.CreateVersion – Create a new version for the input document. Any modification made on the document by the chain will be automatically saved. Increment version if this was specified through the 'snapshot' parameter. Returns the live document (not the version).

SYNTAX

```
Document.CreateVersion [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -increment -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Delete

NAME
Document.Delete – Delete the input document. The previous context input will be restored for the next operation.

SYNTAX

```
Document.Delete [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Fetch

NAME
Document.Fetch – Fetch a document from the repository given its reference (path or UID). The document will become the input of the next operation.

SYNTAX

```
Document.Fetch [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -value -
- -void - [flag] - If void the server will not return the result back

Document.FetchByProperty

NAME
Document.FetchByProperty – For each specified string property value, fetch all documents that match the property and the optional where clause. Matching documents are collected into a list and the returned to the next operation. The operation has no input.

SYNTAX

```
Document.FetchByProperty [options]
```

OPTIONS

- -values -
- -ctx - Can be used to inject context properties in Java properties format
- -property -
- -void - [flag] - If void the server will not return the result back
- -query -

Document.Filter

NAME
Document.Filter – Filter the input list of documents given a condition. The condition can be expressed using 4 parameters: types, facets, lifecycle and condition. If more than one parameter is specified an AND will be used to group conditions.
The 'types' parameter can take a comma separated list of document type: File,Note.
The 'facet' parameter can take a single facet name.
The 'life cycle' parameter takes a name of a life cycle state the document should have.
The 'condition' parameter can take any EL expression.
Returns the list of documents that match the filter condition.

SYNTAX

```
Document.Filter [options] [the input document(s)]
```

OPTIONS

- -class -
- -types -
- -pathStartsWith -
- -ctx - Can be used to inject context properties in Java properties format
- -facet -
- -void - [flag] - If void the server will not return the result back
- -lifecycle -

- -condition -

ARGUMENTS

- the input document(s) - [optional] -null

Document.GetChild

NAME
Document.GetChild – Get a child document given its name. Take as input the parent document and return the child document.

SYNTAX

```
Document.GetChild [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.GetChildren

NAME
Document.GetChildren – Get the children of a document. The list of children will become the input for the next operation

SYNTAX

```
Document.GetChildren [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.GetParent

NAME
Document.GetParent – Get the parent document of the input document. The parent document will become the input for the next operation. You can use the 'type' parameter to specify which parent to select from the document ancestors

SYNTAX

```
Document.GetParent [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -type -

ARGUMENTS

- the input document(s) - [optional] -null

Document.GetPrincipalEmails

NAME

Document.GetPrincipalEmails – Fetch the principal emails that have a given permission on the input document and then set them in the context under the given key variable name. The operation returns the input document. You can later use the list of principals set by this operation on the context from another operation. The 'key' argument represents the variable name and the 'permission' argument the permission to check. If the 'ignore groups' argument is false then groups are recursively resolved, extracting user members of these groups. Be **warned** that this may be a very consuming operation.
Note that:

- groups are not included
- the list pushed into the context is a string list of emails.

SYNTAX

```
Document.GetPrincipalEmails [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -ignore groups -
- -variable name -
- -void - [flag] - If void the server will not return the result back
- -permission -

ARGUMENTS

- the input document(s) - [optional] -null

Document.GetUsersAndGroups

NAME
Document.GetUsersAndGroups – Fetch the users and groups that have a given permission on the input document and then set them in the context under the given key variable name. The operation returns the input document. You can later use the list of identifiers set by this operation on the context from another operation. The 'key' argument represents the variable name and the 'permission' argument the permission to check. If the 'ignore groups' argument is false then groups will be part of the result. If the 'resolve groups' argument is true then groups are recursively resolved, adding user members of these groups in place of them. Be warned that this may be a very consuming operation. If the 'prefix identifiers' argument is true, then user identifiers are prefixed by 'user:' and groups identifiers are prefixed by 'group:'.

SYNTAX

```
Document.GetUsersAndGroups [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -ignore groups -
- -resolve groups -
- -variable name -
- -void - [flag] - If void the server will not return the result back
- -permission -
- -prefix identifiers -

ARGUMENTS

- the input document(s) - [optional] -null

Document.Lock

NAME
Document.Lock – Lock the input document in the name of the given 'owner'. The lock owner is an username and identifies the user that owns the lock on the document. If the owner is not specified, the current user will be used as the owner. Returns back the locked document.

SYNTAX

```
Document.Lock [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -owner -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Move

NAME
Document.Move – Move the input document into the target folder.

SYNTAX

```
Document.Move [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -target -
- -name -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.MultiPublish

NAME
Document.MultiPublish – Publish the input document(s) into several target sections. The target is evaluated to a document list (can be a path, UID or EL expression). Existing proxy is overriden if the override attribute is set. Returns a list with the created proxies.

SYNTAX

```
Document.MultiPublish [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -target -
- -void - [flag] - If void the server will not return the result back
- -override -

ARGUMENTS

- the input document(s) - [optional] -null

Document.Pop

NAME
Document.Pop – Restore the last saved input document in the context input stack. This operation must be used only if a PUSH operation was previously made. Return the last *pushed* document.

SYNTAX

```
Document.Pop [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Document.PopList

NAME
Document.PopList – Restore the last saved input document list in the context input stack

SYNTAX

```
Document.PopList [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Document.Publish

NAME
Document.Publish – Publish the input document into the target section. Existing proxy is overriden if the override attribute is set. Return the created proxy.

SYNTAX

```
Document.Publish [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -target -
- -void - [flag] - If void the server will not return the result back
- -override -

ARGUMENTS

- the input document(s) - [optional] -null

Document.Pull

NAME
Document.Pull – Restore the first saved input document in the context input stack. This operation must be used only if a PUSH operation was previously made. Return the first *pushed* document.

SYNTAX

```
Document.Pull [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Document.PullList

NAME
Document.PullList – Restore the first saved input document list in the context input stack

SYNTAX

```
Document.PullList [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Document.Push

NAME
Document.Push – Push the input document on the context stack. The document can be restored later as the input using the corrresponding pop operation. Returns the input document.

SYNTAX

```
Document.Push [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.PushList

NAME
Document.PushList – Push the input document list on the context stack. The document list can be restored later as the input using the corrresponding pop operation. Returns the input document list.

SYNTAX

```
Document.PushList [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Query

NAME
Document.Query – Perform a query on the repository. The query result will become the input for the next operation.

SYNTAX

```
Document.Query [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -language -
- -void - [flag] - If void the server will not return the result back
- -query -

Document.Reload

NAME
Document.Reload – Reload the input document from the repository. Any previous modification made by the chain on this document will be lost if these modifications were not saved. Return the reloaded document.

SYNTAX

```
Document.Reload [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.RemoveACL

NAME
Document.RemoveACL – Remove a named Acces Control List from the input document(s). Returns the document(s).

SYNTAX

```
Document.RemoveACL [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -acl -

ARGUMENTS

- the input document(s) - [optional] -null

Document.RemoveProperty

NAME
Document.RemoveProperty – Remove the given property of the input document(s) as specified by the 'xpath' parameter. If the property points to a list then clear the list. Removing a property means setting it to null. Return the document(s).

SYNTAX

```
Document.RemoveProperty [options] [the input document(s)]
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Document.Save

NAME
Document.Save – Save in the repository any modification that was done on the input document. Returns the saved document.

SYNTAX

```
Document.Save [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.SaveSession

NAME
Document.SaveSession – Commit any changes made by the operation on the documents. This can be used to explicitly commit changes. This operation can be executed on any type of input. The input of this operation will be preserved as the input for the next operation in the chain.

SYNTAX

```
Document.SaveSession [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

Document.SetACE

NAME
Document.SetACE – Set Acces Control Entry on the input document(s). Returns the document(s).

SYNTAX

```
Document.SetACE [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -grant -
- -overwrite -
- -void - [flag] - If void the server will not return the result back
- -user -
- -acl -
- -permission -

ARGUMENTS

- the input document(s) - [optional] -null

Document.SetLifeCycle

NAME
Document.SetLifeCycle – Follow the given transition on the input document life cycle state

SYNTAX

```
Document.SetLifeCycle [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -value -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.SetProperty

NAME
Document.SetProperty – Set a single property value on the input document. The property is specified using its XPath. The document is automatically saved if 'save' parameter is true. If you unset the 'save' you need to save it later using Save Document operation. Return the modified document.

SYNTAX

```
Document.SetProperty [options] [the input document(s)]
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -value -
- -void - [flag] - If void the server will not return the result back
- -xpath -

ARGUMENTS

- the input document(s) - [optional] -null

Document.Unlock

NAME
Document.Unlock – Unlock the input document. The unlock will be executed in the name of the current user. An user can unlock a document only if has the UNLOCK permission granted on the document or if it the same user as the one that locked the document. Return the unlocked document

SYNTAX

```
Document.Unlock [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Document.Update

NAME
Document.Update – Set multiple properties on the input document. The properties are specified as *key=value* pairs separated by a new line. The key used for a property is the property XPath. To specify multi-line values you can use a \ character followed by a new line.
Example:
`dc:title=The Document Title`
`dc:description=foo bar`
Returns back the updated document.

SYNTAX

```
Document.Update [options] [the input document(s)]
```

OPTIONS

- -save -
- -ctx - Can be used to inject context properties in Java properties format
- -properties -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Notification.SendEvent

NAME
Notification.SendEvent – Send a Nuxeo event.

SYNTAX

```
Notification.SendEvent [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -name -
- -void - [flag] - If void the server will not return the result back

Notification.SendMail

NAME
Notification.SendMail – Send an email using the input document to the specified recipients. You can use the asHTML parameter to specify whether you message is in HTML format or in plain text. Also you can attach any blob on the current document to the message by using the comma separated list of XPath expressions 'files'. If your XPath points to a blob list all blobs in the list will be attached. Return back the input document(s).

SYNTAX

```
Notification.SendMail [options] [the input document(s)]
```

OPTIONS

- -viewId -
- -message -
- -ctx - Can be used to inject context properties in Java properties format
- -from -
- -files -
- -void - [flag] - If void the server will not return the result back
- -to -
- -subject -
- -asHTML -

ARGUMENTS

- the input document(s) - [optional] -null

Relations.CreateRelation

NAME
Relations.CreateRelation – Create a relation between 2 documents. The subject of the relation will be the input of the operation and the object of the relation will be retrieved from the context using the 'object' field. The 'predicate' field specify the relation predicate. Return back the subject document.

SYNTAX

```
Relations.CreateRelation [options] [the input document(s)]
```

OPTIONS

- -object -
- -ctx - Can be used to inject context properties in Java properties format
- -predicate -
- -void - [flag] - If void the server will not return the result back

ARGUMENTS

- the input document(s) - [optional] -null

Relations.GetRelations

NAME
Relations.GetRelations – Get the relations for the input document. The 'outgoing' parameter ca be used to specify whether outgoing or incoming relations should be returned. Returns a document list.

SYNTAX

```
Relations.GetRelations [options] [the input document(s)]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -predicate -
- -void - [flag] - If void the server will not return the result back
- -outgoing -

ARGUMENTS

- the input document(s) - [optional] -null

Workflow.CreateTask

NAME
Workflow.CreateTask – Enable to create a task bound to the document.

Directive, **comment** and **due date** will be displayed in the task list of the user. In **accept operation chain** and **reject operation chain** fields, you can put the operation chain ID of your choice among the one you contributed. Those operations will be executed when the user validates the task, depending on whether he accepts or rejects the task. You have to specify a variable name (the **key for ...** parameter) to resolve target users and groups to which the task will be assigned. You can use Get Users and Groups to update a context variable with some users and groups. If you check **create one task per actor**, each of the actors will have a task to achieve, versus "the first who achieve the task makes it disappear for the others".

SYNTAX

```
Workflow.CreateTask [options] [the input document(s)]
```

OPTIONS

- -variable name for actors prefixed ids -
- -reject operation chain -
- -ctx - Can be used to inject context properties in Java properties format
- -directive -
- -create one task per actor -
- -accept operation chain -
- -additional list of actors prefixed ids -
- -due date -
- -void - [flag] - If void the server will not return the result back
- -comment -
- -task name -

ARGUMENTS

- the input document(s) - [optional] -null

Workflow.GetTask

NAME
Workflow.GetTask – List tasks assigned to this user or one of its group.Task properties are serialized using JSON and returned in a Blob.

SYNTAX

```
Workflow.GetTask [options]
```

OPTIONS

- -ctx - Can be used to inject context properties in Java properties format
- -void - [flag] - If void the server will not return the result back

print

NAME
print – Print operation(s) definition

SYNTAX

```
print [options] [operation]
```

OPTIONS

- -p - The password if any.
- -u - The username if any.
- -out - An optional file to save the operation definition into. If not used the definition will be printed on stdout.

ARGUMENTS

- operation - [optional] -The operation to print.

Related pages

Nuxeo Shell (Nuxeo Installation and Administration - 5.8)

Built-in Commands (Nuxeo Installation and Administration - 5.8)

Nuxeo Automation Commands (Nuxeo Installation and Administration - 5.8)

Filesystem Commands (Nuxeo Installation and Administration - 5.8)

Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)

Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)

Shell Commands (Nuxeo Platform Developer Documentation - 5.8)

Shell Features (Nuxeo Platform Developer Documentation - 5.8)

Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)

Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)

Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)

Configuration Commands (Nuxeo Installation and Administration - 5.8)

- ## Configuration Commands

Namespace: *config*
Commands for configuring the shell.

Index:

background

NAME
background – Modify the background color used by the shell. This command is available only in UI mode.

SYNTAX

```
background
```

color

NAME
color – Modify the foreground color used by the shell. This command is available only in UI mode.

SYNTAX

```
color
```

font

NAME
font – Modify the font used by the shell. This command is available only in UI mode.

SYNTAX

```
font
```

settings

NAME
settings – Print or modify the shell settings.

SYNTAX

```
settings [options] [name] [value]
```

OPTIONS

- -reset - [flag] - Reset settings to their defaults. Need to restart shell.

ARGUMENTS

- name - [optional] -The variable to print or set.
- value - [optional] -The variable value to set.

theme

NAME
theme – Modify the theme used by the shell. This command is available only in UI mode.

SYNTAX

```
theme [name]
```

ARGUMENTS

- name - [optional] -The theme name to set. If not specified the current theme is printed.

Related pages

Nuxeo Shell (Nuxeo Installation and Administration - 5.8)
Nuxeo Shell Command Index (Nuxeo Installation and Administration - 5.8)
Built-in Commands (Nuxeo Installation and Administration - 5.8)
Nuxeo Automation Commands (Nuxeo Installation and Administration - 5.8)
Filesystem Commands (Nuxeo Installation and Administration - 5.8)
Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)
Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)
Shell Commands (Nuxeo Platform Developer Documentation - 5.8)
Shell Features (Nuxeo Platform Developer Documentation - 5.8)
Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)
Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)
Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)
Configuration Commands (Nuxeo Installation and Administration - 5.8)

• Nuxeo Shell Batch Mode

As said in the Overview section there are three modes to execute batch commands:

1. Run commands from a file
2. Run commands from standard input.
3. Run commands specified on the command line - this is a convenient way to run a short list of commands.

Running Commands From a File

In this section

To run commands from a file you should use the **-f** parameter to specify a file containing commands when launching Nuxeo Shell.

Example:

```
java -cp nuxeo-shell.jar org.nuxeo.shell.Main -f my_batch_file
```

Where **my_batch_file** is a file containing the commands to execute - each command on one line. Empty lines and lines begining with **#** are ignored. The **#** character can be use to add comments to a batch file.

Here is an example of a batch file:

```
# connect to local server using the Administrator account.
connect -u Administrator -p Administrator
http://localhost:8080/nuxeo/site/automation

# we are now in the repository root. Go to /default-domain/workspaces
cd /default-domain/workspaces

# list the content of the workspaces root - as document UIDs
ls -uid
```

If you want to span a command on multiple lines (you may want this for improved readability in case of long commands) you can end the line with a *** character (*make sure** you don't have a space after **). In that case the command will continue on the next line, and so on until no more line ending ** is found or the end of file is reached.

Example:

```
# connect to local server using the Administrator account.
connect -u Administrator -p Administrator
http://localhost:8080/nuxeo/site/automation

# get all workspaces in the repository
query "SELECT * FROM Document WHERE ecm:primaryType='Workspace' \
      ORDER BY dc:title DESC"
```

Running Commands From Standard Input

If you want to run batch commands from the terminal standard input you can use the **-** option when launching the Nuxeo shell.
The format of the commands is the same as the one described when running commands from a file.

Here is an example which will run the commands from my_batch_file file by using the Unix **cat** application and pipes:

```
cat my_batch_file | java -cp nuxeo-shell.jar org.nuxeo.shell.Main -
```

Running Batch Commands from the Command Line

If you just run a few short commands you can specify them directly in the command line of the Nuxeo Shell.

Example:

```
java -cp nuxeo-shell.jar org.nuxeo.shell.Main -e "connect -u Administrator -p
Administrator http://localhost:8080/nuxeo/site/automation; ls"
```

Note that commands are separated using a semicolon character.

Limitations
You cannot run that way commands that contains illegal characters and needs to be escaped.

Related pages

- Nuxeo Shell (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Command Index (Nuxeo Installation and Administration - 5.8)
- Built-in Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Automation Commands (Nuxeo Installation and Administration - 5.8)
- Filesystem Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Server Commands (Nuxeo Installation and Administration - 5.8)
- Nuxeo Shell Batch Mode (Nuxeo Installation and Administration - 5.8)
- Shell Commands (Nuxeo Platform Developer Documentation - 5.8)
- Shell Features (Nuxeo Platform Developer Documentation - 5.8)
- Extending The Shell (Nuxeo Platform Developer Documentation - 5.8)
- Shell Documentation (Nuxeo Platform Developer Documentation - 5.8)
- Shell Namespaces (Nuxeo Platform Developer Documentation - 5.8)
- Configuration Commands (Nuxeo Installation and Administration - 5.8)

Admin Center overview

The Admin Center is a space within the Nuxeo Platform that provides administrative services, such as server and application usage summary information, as well as access to upgrades, patches, the Nuxeo Marketplace, and Nuxeo Studio projects. It is available to administrators and power users, whose access is limited to some features.

The Admin Center offers access to different kinds of information about your Nuxeo instance. Depending the modules and additional packages you have installed on the Platform, you may have more or less information, i.e. more or less tabs.

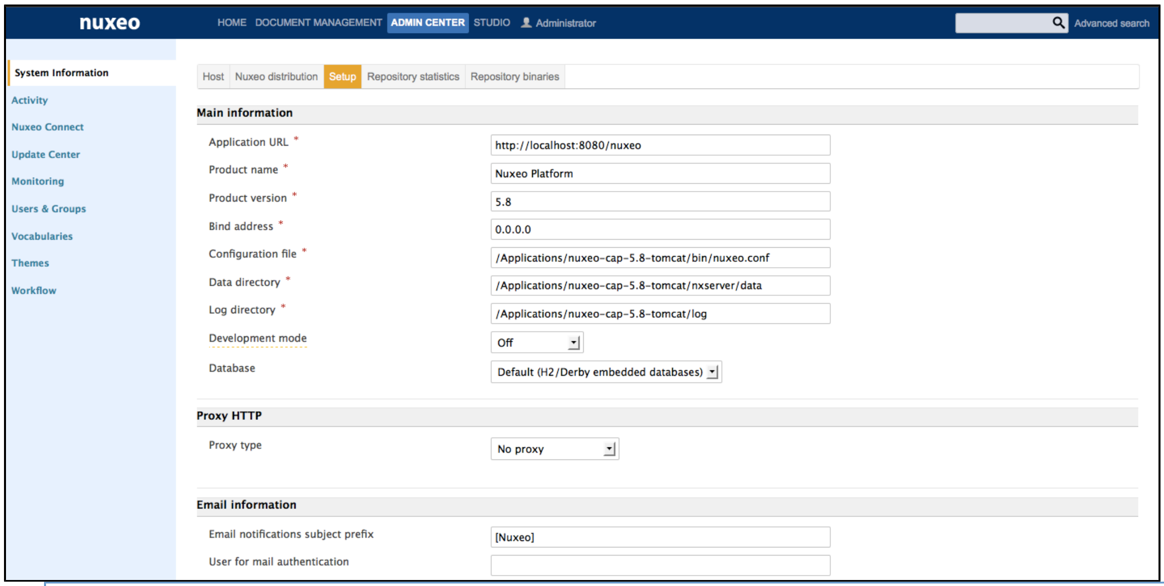
Installing Marketplace packages can add new tabs in the Admin Center. In that case, the new Admin Center feature is described in the addon's documentation.

Admin Center Default Tabs

The default tabs available in the Admin Center are listed below. These tabs are available whatever the modules or addons you installed on the Platform.

System Information

This section of the Admin Center provides information about the server the Nuxeo Platform is installed on, about your instance configuration.


nuxeo
HOME DOCUMENT MANAGEMENT ADMIN CENTER STUDIO Administrator
Advanced search
System Information
Activity
Nuxeo Connect
Update Center
Monitoring
Users & Groups
Vocabularies
Themes
Workflow
Host Nuxeo distribution Setup Repository statistics Repository binaries

Main information

Application URL *	http://localhost:8080/nuxeo
Product name *	Nuxeo Platform
Product version *	5.8
Bind address *	0.0.0.0
Configuration file *	/Applications/nuxeo-cap-5.8-tomcat/bin/nuxeo.conf
Data directory *	/Applications/nuxeo-cap-5.8-tomcat/nxserver/data
Log directory *	/Applications/nuxeo-cap-5.8-tomcat/log
Development mode	Off
Database	Default (H2/Derby embedded databases)

Proxy HTTP

Proxy type	No proxy

Email information

Email notifications subject prefix	[Nuxeo]
User for mail authentication	

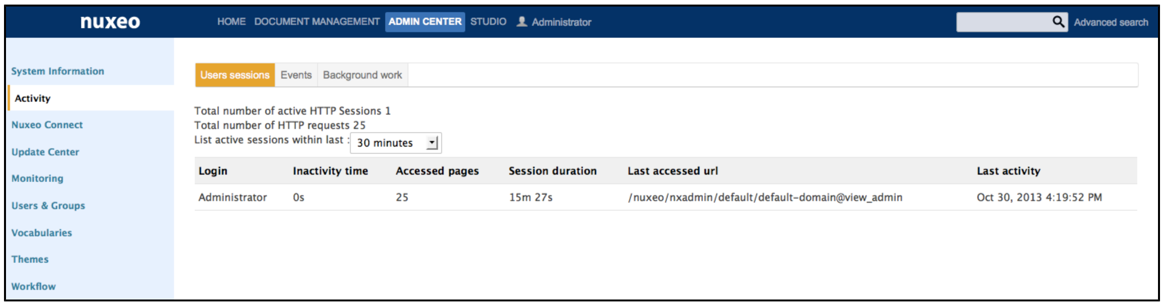
In this section

The different sub-tabs are:

- **Host**: provides information about the computer on which the Nuxeo Platform is installed. This is also where you can easily restart Nuxeo, using the **Restart server** button.
- **Nuxeo distribution**: provides information on the version of the Nuxeo Platform you are using.
- **Setup**: enables you to change your application's configuration, such as where the logs are stored, or the default port. Some examples are available on the Configuration examples and Recommended configurations pages. This tab enables you to perform the same configuration modification as you can do manually by editing the nuxeo.conf file.
- **Repository statistics**: provides some statistics about the content of your application (how many documents, how many versions, size of the biggest folder...).
- **Repository binaries**: computes statistics on the binaries and enables you to delete the unused ones (deleted from the UI but still stored on the server).

Activity

This section of the Admin Center enables administrators to have information and statistics on how the application is used.



Total number of active HTTP Sessions 1
Total number of HTTP requests 25
List active sessions within last : 30 minutes

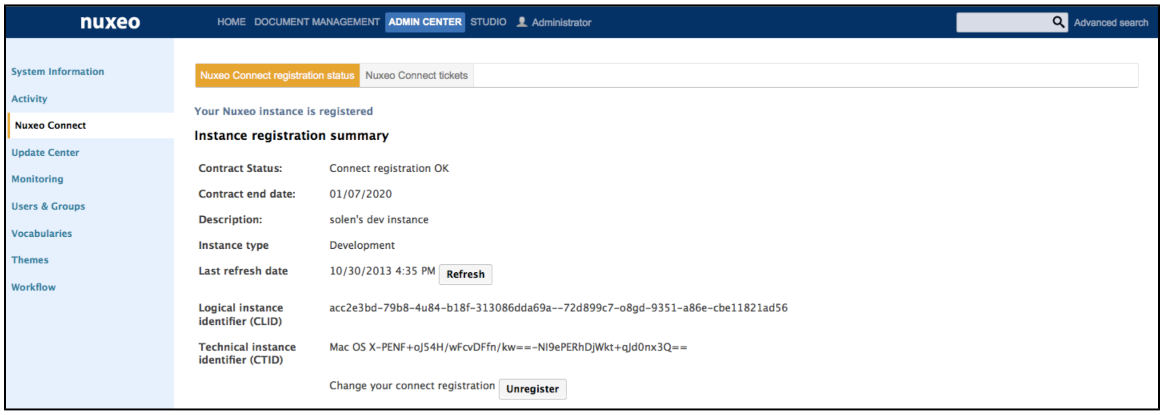
Login	Inactivity time	Accessed pages	Session duration	Last accessed url	Last activity
Administrator	0s	25	15m 27s	/nuxeo/nxadmin/default/default-domain@view_admin	Oct 30, 2013 4:19:52 PM

The default Activity subtabs are:

- **Users sessions**: provides information on who is logged in to the Platform.
- **Events**: Lists the events that occurred on the platform. You can filter this list to only get the events from a specific user or only events from a specific category.
- **Background work**: provides information on the asynchronous tasks performed by the server, such as video conversion when a video is imported.

Nuxeo Connect

You can connect your Nuxeo instance with your Nuxeo Connect account. This enables you to manage your Support tickets from your Nuxeo application, to install hotfixes and to connect your Nuxeo instance to the Nuxeo Marketplace and use any addon you want.



Your Nuxeo instance is registered

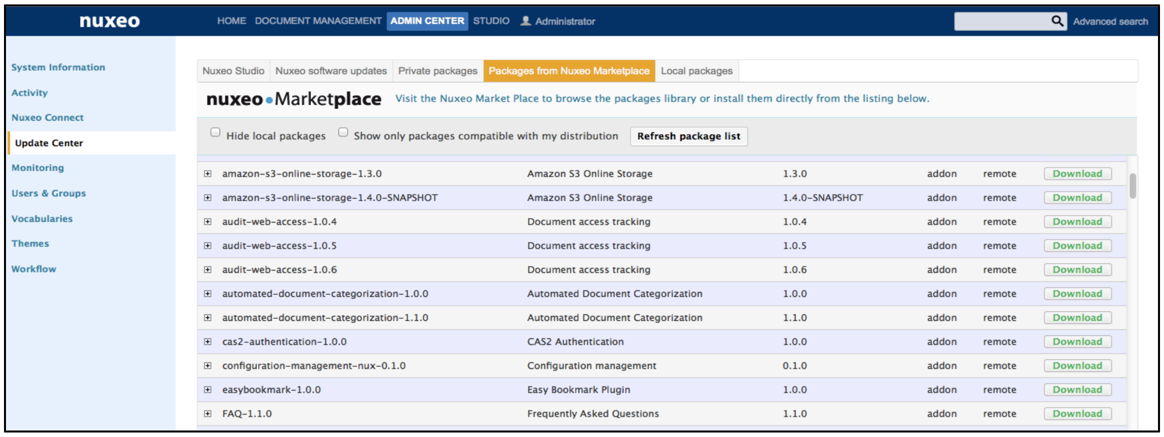
Instance registration summary

Contract Status:	Connect registration OK
Contract end date:	01/07/2020
Description:	solen's dev instance
Instance type	Development
Last refresh date	10/30/2013 4:35 PM
Logical instance identifier (CLID)	acc2e3bd-79b8-4u84-b18f-313086dda69a--72d899c7-o8gd-9351-a86e-cbe11821ad56
Technical instance identifier (CTID)	Mac OS X-PENF+oJ54H/wFcvDFfn/kw==-NI9ePERhDjWkt+qJd0nx3Q==
	Change your connect registration

- The **Nuxeo Connect registration status** tab is where you can register your instance, if you haven't done it earlier from the setup wizard. When you have already registered, it displays your contract's and instance's information.
- The **Nuxeo Connect tickets** tab will display the list of JIRA issues linked to your contract. This feature is not yet available, but you can see all your JIRA issues from the My support cases view on the Nuxeo Connect website.

Update Center

This section of the Admin Center provides all the updates you can need (updates and hotfixes, Nuxeo Marketplace addons, direct access to your Nuxeo Studio customizations, local packages).



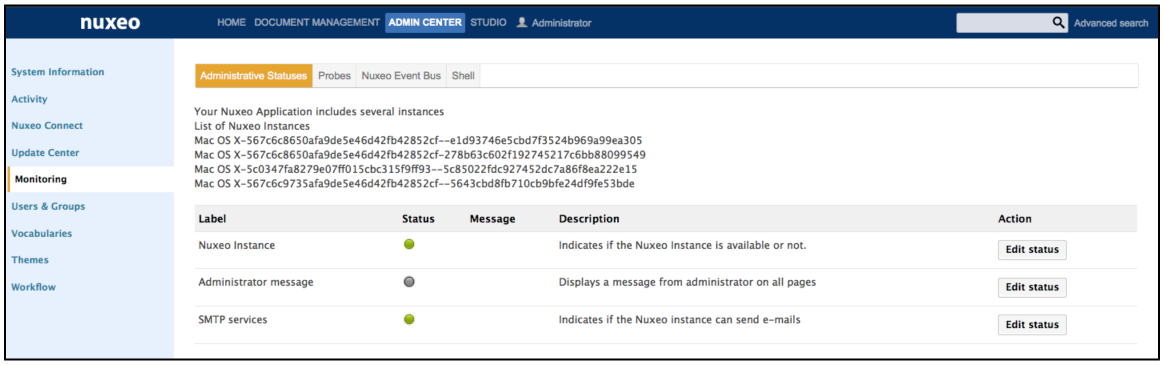
Visit the Nuxeo Market Place to browse the packages library or install them directly from the listing below.

amazon-s3-online-storage-1.3.0	Amazon S3 Online Storage	1.3.0	addon	remote	Download
amazon-s3-online-storage-1.4.0-SNAPSHOT	Amazon S3 Online Storage	1.4.0-SNAPSHOT	addon	remote	Download
audit-web-access-1.0.4	Document access tracking	1.0.4	addon	remote	Download
audit-web-access-1.0.5	Document access tracking	1.0.5	addon	remote	Download
audit-web-access-1.0.6	Document access tracking	1.0.6	addon	remote	Download
automated-document-categorization-1.0.0	Automated Document Categorization	1.0.0	addon	remote	Download
automated-document-categorization-1.1.0	Automated Document Categorization	1.1.0	addon	remote	Download
cas2-authentication-1.0.0	CAS2 Authentication	1.0.0	addon	remote	Download
configuration-management-nux-0.1.0	Configuration management	0.1.0	addon	remote	Download
easybookmark-1.0.0	Easy Bookmark Plugin	1.0.0	addon	remote	Download
FAQ-1.1.0	Frequently Asked Questions	1.1.0	addon	remote	Download

- The **Nuxeo software updates** tab displays the hotfixes available for your Nuxeo version and the updates for the packages you have installed.
- The **Nuxeo Studio** tab enables you to update your application with your Nuxeo Studio customizations.
- The **Private packages** tab displays the lists of packages specific to your project, hence private: the Studio customizations and possible packaged customizations done outside Studio.
- The **Packages from Nuxeo Marketplace** displays the list of all available packages.
- The **Local packages** tab displays the packages (hotfixes, addons, Studio customization) you have downloaded and that are either ready to be installed or already installed.

Monitoring

This section of the Admin Center enables administrators to monitor some technical information and display messages to users.



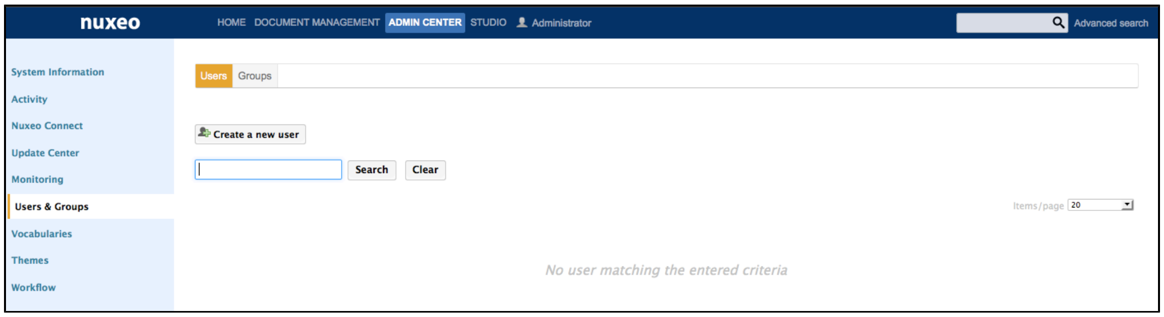
Your Nuxeo Application includes several instances
List of Nuxeo Instances
Mac OS X-567c6c8650afa9de5e46d42fb42852cf--e1d93746e5cbd7f3524b969a99ea305
Mac OS X-567c6c8650afa9de5e46d42fb42852cf-278b63c602f192745217c6bb88099549
Mac OS X-5c0347fa8279e07ff015cbc315f9ff93--5c85022fdc927452dc7a86f8ea222e15
Mac OS X-567c6c9735afa9de5e46d42fb42852cf--5643cbd8fb710cb9bfe24df9fe53bde

Label	Status	Message	Description	Action
Nuxeo Instance	●		Indicates if the Nuxeo Instance is available or not.	Edit status
Administrator message	●		Displays a message from administrator on all pages	Edit status
SMTP services	●		Indicates if the Nuxeo instance can send e-mails	Edit status

- The **Administrative Statuses** tab is the place from which you can display messages to the application's users, make the application unavailable or to make SMTP services unavailable.
- The **Probes** tab enables you to check the access to the SQL repositories, LDAP directory and the local Nuxeo instance.
- The **Nuxeo Event Bus** displays statistics about the processing of listeners.
- The **Shell** tab enables users to use the Nuxeo Shell.

Users & Groups

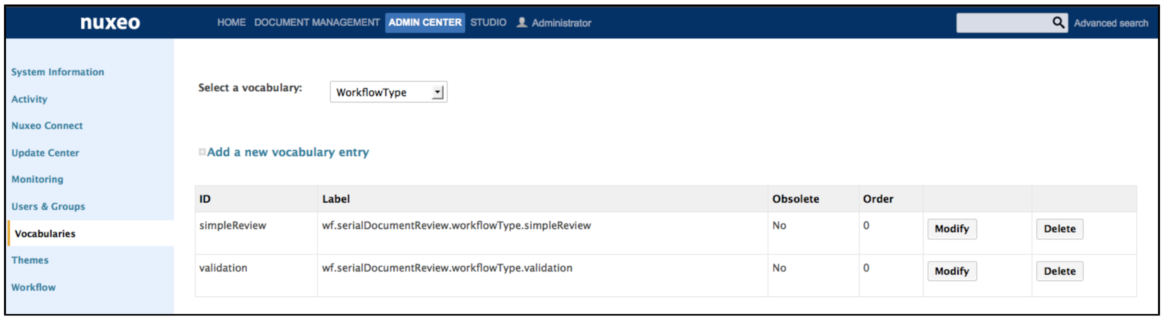
This section enables administrators to create, edit, delete users and groups of users.



No user matching the entered criteria

Vocabularies

From this section, administrators can customize and adapt the values displayed in lists in the application.

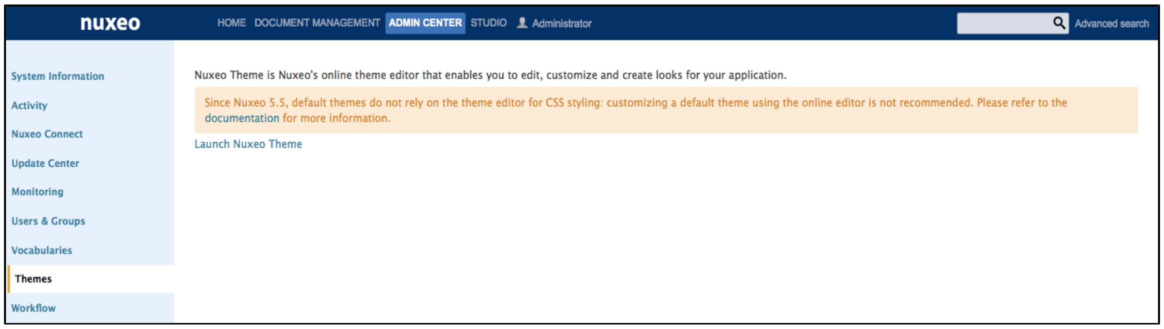


Select a vocabulary: WorkflowType

ID	Label	Obsolete	Order		
simpleReview	wf.serialDocumentReview.workflowType.simpleReview	No	0	Modify	Delete
validation	wf.serialDocumentReview.workflowType.validation	No	0	Modify	Delete

Themes

This section gives administrators access to Nuxeo Themes, that enables them to create new themes for the application.

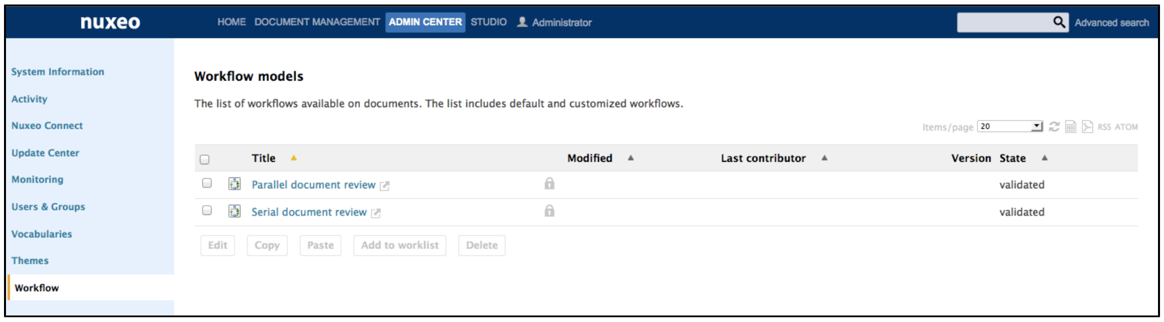


Nuxeo Theme is Nuxeo's online theme editor that enables you to edit, customize and create looks for your application.

Since Nuxeo 5.5, default themes do not rely on the theme editor for CSS styling: customizing a default theme using the online editor is not recommended. Please refer to the documentation for more information.

Workflow

This section enables administrators to manage the different workflows available (including your customized workflows from Studio) and especially to define for which users the workflow is available.



Workflow models

The list of workflows available on documents. The list includes default and customized workflows.

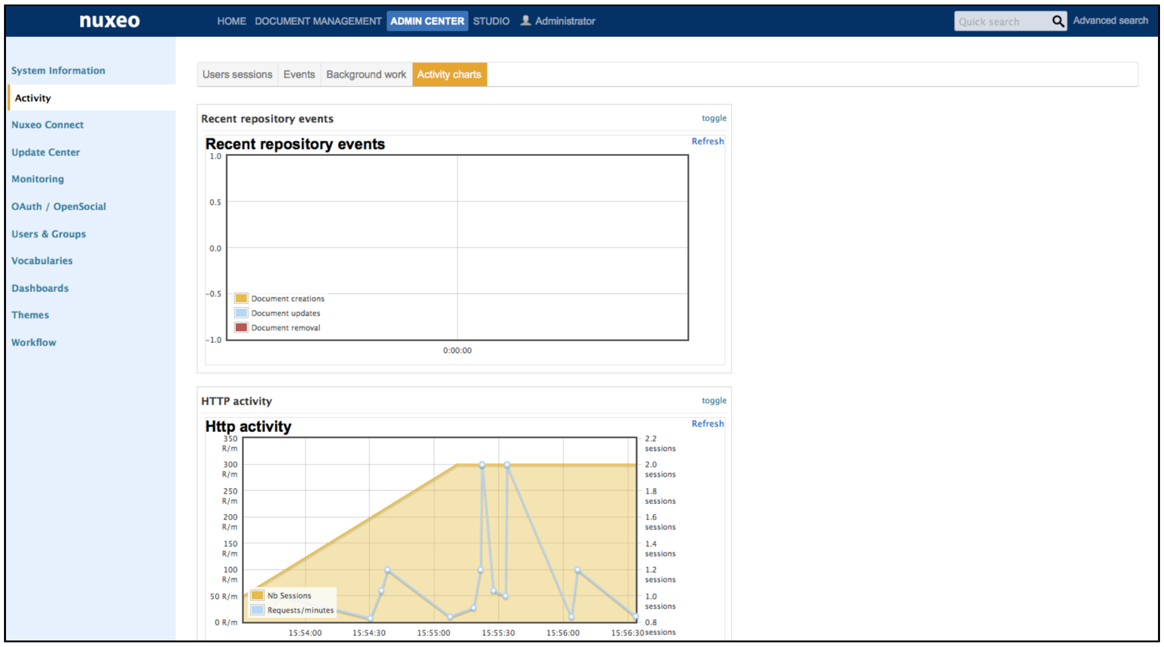
	Title	Modified	Last contributor	Version	State
	Parallel document review				validated
	Serial document review				validated

Document Management Module Tabs

The Document Management module adds the tabs below.

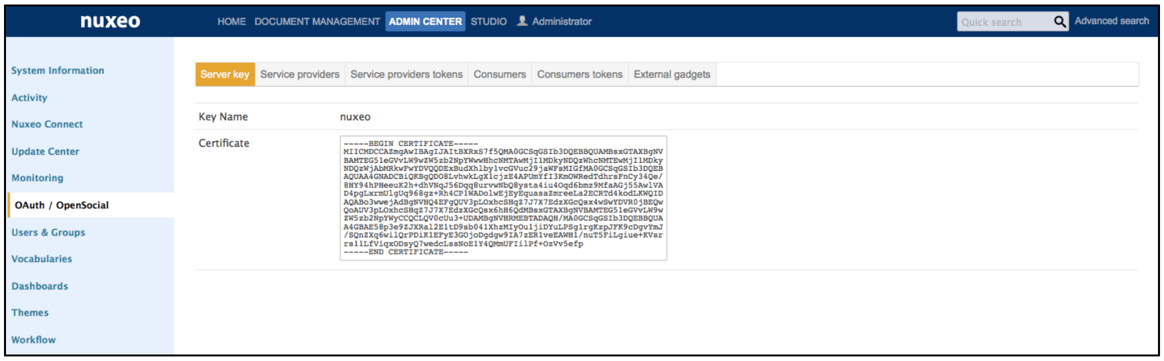
New Activity Tab: Activity Charts

Installing the Document Management module on the Nuxeo Platform adds a new tab in the **Activity** section of the Admin Center, called **Activ ity charts**. As its name stands it, it displays graphical charts showing how the activity on the server evolved.



OAuth / Open Social

This section of the Admin Center enables administrators to manage the authentication with other applications using OAuth protocol and to ad d new gadgets.



Key Name nuxeo

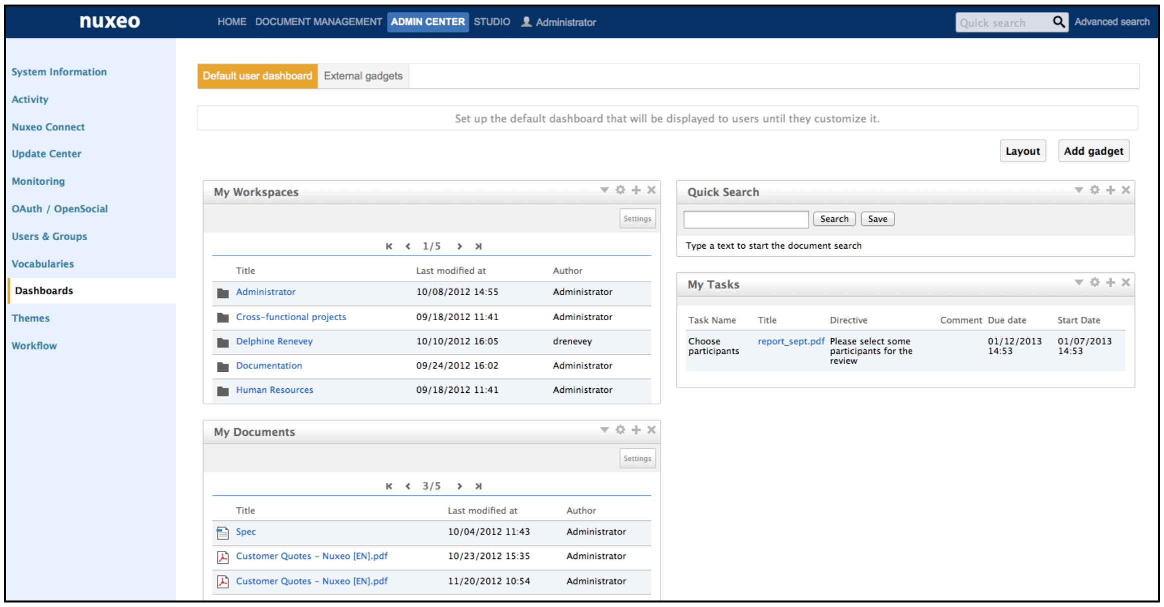
Certificate

```
-----BEGIN CERTIFICATE-----
MIICMDCCAZmgAwIBAgIJAItBXRxS7f5QMA0GCSqGSIb3DQEBBQUAMBsxGTAXBgNV
BAMTEG5leGVvLW9wZW5zb2NpYWwwHhcNMTAwMjI1MDkyNDQzWhcNMTEwMjI1MDky
NDQzWjAbMRkwFwYDVQQDExBudXhlby1vcGVuc29jaWFsMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQDO8LvhwkLgX1cjzE4APUmYfI3KmOWRedTdhrsFnCy34Qe/
8HY94hPHeeuK2h+dhVNqJ56Dqq8urvwNbQ8ysta4iu4Oqd6bmz9MfaAGj55AwlVA
D4pgLxrmUlgUq968gz+Rh4CPlWADolwEjEyEquasaZmreeLa2ECRTd4kodLKWQID
AQABo3wwejAdBgNVHQ4EFgQUV3pLOxhcSHqZ7J7X7EdzXGcQsx4wSwYDVR0jBEQw
QoAUV3pLOxhcSHqZ7J7X7EdzXGcQsx6hH6QdMBsxGTAXBgNVBAMTEG5leGVvLW9w
ZW5zb2NpYWyCCQCLQV0cUu3+UDAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUA
A4GBAE58p3e9ZJXRal2E1tD9sb041XhzMIyOuljiDYuLPSg1rgKzpJFK9cDgvYmJ
/SQnZXq6wilQrPDiKlEFyE3GOjoDgdgw9IA7zERlveEAWHl/nuT5FiLgiue+KVsr
rs1lLfViqxODsyQ7wedcLssNoE1Y4QMmUFIilPf+OzVv5efp
-----END CERTIFICATE-----
```

- The **Server key** tabs shows the Nuxeo certificate that is required by other applications.
- The **Services providers** and **Services providers tokens** tabs enable to declare external applications to which the Nuxeo Platform will connect.
- The **Consumers** and **Consumers tokens** tabs enable to declare external applications that will connect to the Nuxeo Platform.

Dashboards

This section enables administrators to define the dashboard displayed by default to users and to add new gadgets.



Set up the default dashboard that will be displayed to users until they customize it.

My Workspaces

1/5

Title	Last modified at	Author
Administrator	10/08/2012 14:55	Administrator
Cross-functional projects	09/18/2012 11:41	Administrator
Delphine Renevey	10/10/2012 16:05	drenevey
Documentation	09/24/2012 16:02	Administrator
Human Resources	09/18/2012 11:41	Administrator

Quick Search

Type a text to start the document search

My Tasks

Task Name	Title	Directive	Comment	Due date	Start Date
Choose participants	report_sept.pdf	Please select some participants for the review		01/12/2013 14:53	01/07/2013 14:53

My Documents

3/5

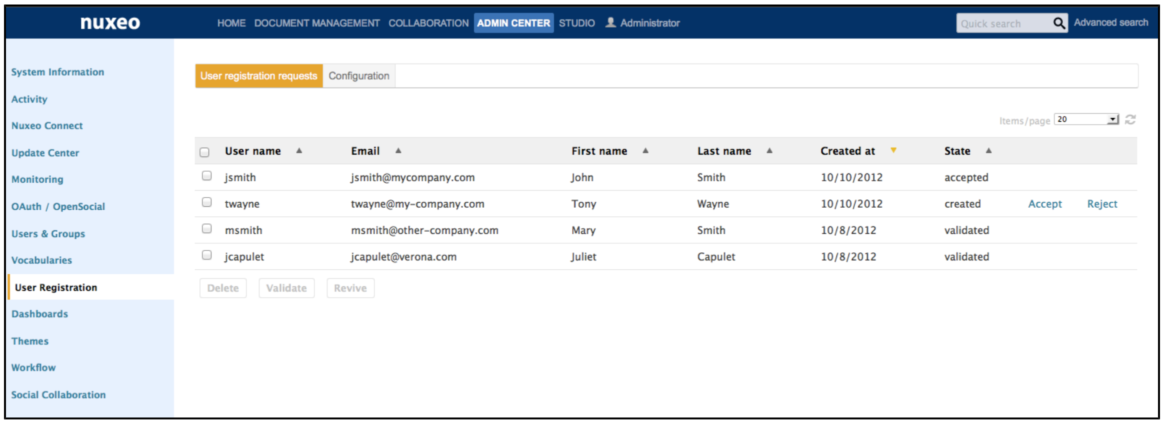
Title	Last modified at	Author
Spec	10/04/2012 11:43	Administrator
Customer Quotes - Nuxeo [EN].pdf	10/23/2012 15:35	Administrator
Customer Quotes - Nuxeo [EN].pdf	11/20/2012 10:54	Administrator

Social Collaboration Module Tabs

The Social Collaboration module adds the tabs below.

Users Registrations

From this section, administrators can manage invitations to workspaces.



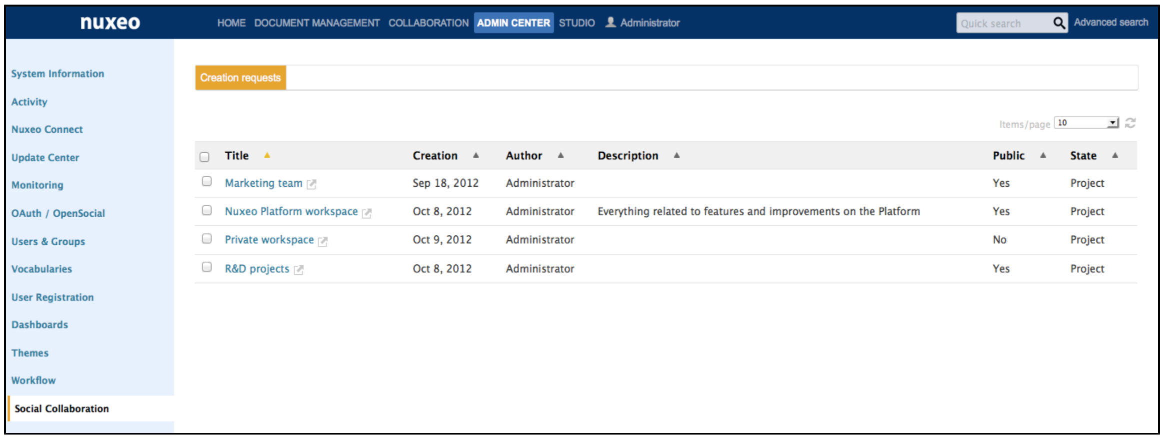
	User name	Email	First name	Last name	Created at	State		
	jsmith	jsmith@mycompany.com	John	Smith	10/10/2012	accepted		
	twayne	twayne@my-company.com	Tony	Wayne	10/10/2012	created	Accept	Reject
	msmith	msmith@other-company.com	Mary	Smith	10/8/2012	validated		
	jcapulet	jcapulet@verona.com	Juliet	Capulet	10/8/2012	validated		

- From the **User registration requests** tab, they can validate, refuse or revive requests to invite external people to access the Platform.
- From the **Configuration** tab, they can define how the invitation system should behave (limiting invitations to users that already have a user account for instance).

This tab is also available without the Social Collaboration module when you install the Nuxeo Platform User Registration addon.

Social Collaboration

This section enables administrators to manage the creation of new social workspaces.



	Title	Creation	Author	Description	Public	State
	Marketing team	Sep 18, 2012	Administrator		Yes	Project
	Nuxeo Platform workspace	Oct 8, 2012	Administrator	Everything related to features and improvements on the Platform	Yes	Project
	Private workspace	Oct 9, 2012	Administrator		No	Project
	R&D projects	Oct 8, 2012	Administrator		Yes	Project

Registering your Nuxeo Instance

Registering your Nuxeo application with Nuxeo Connect will give you access to a wide range of services, such as:

- the Update Center, so that you can easily install patches and bug fixes,
- a global view of your open support tickets,
- the Nuxeo Marketplace, the app store for the Nuxeo community, that provides an easy and powerful way to add features and plugins to your Nuxeo application,
- an interface to update of your Nuxeo application with your Studio configuration, including hot redeploy.

The registration process only copies a file on your file system. This enables the Nuxeo Connect portal to identify the instance among all the registered instances. You can register multiple instances.

On this page

How to Register

To be able to register, you need to have a Nuxeo Connect account.

Registration can be done during the installation steps, using the configuration wizard or at anytime later, through the Admin Center. Registration doesn't require an Internet access. If your server cannot connect to the Internet, follow the offline registration steps. Otherwise, follow the online registration steps.

For development instances on which you may need to remove your data, you may need to re-register your instance.

Creating your Nuxeo Connect Account

If you already have an account on Nuxeo Connect, either because you are a Nuxeo customer, or because you created a trial account, you can continue to the Registering online step. If not, follow those steps to get credentials to the Nuxeo Connect portal.

To subscribe to a Connect trial:

1. Go to the Connect trial registration form.
2. Fill in the form. Provide a valid email address or else registration will not be completed.
3. Confirm registration by clicking on the link sent to the email address in the previous step.
4. If it is not already done, download the last published version of the Nuxeo Platform.

Registering Online

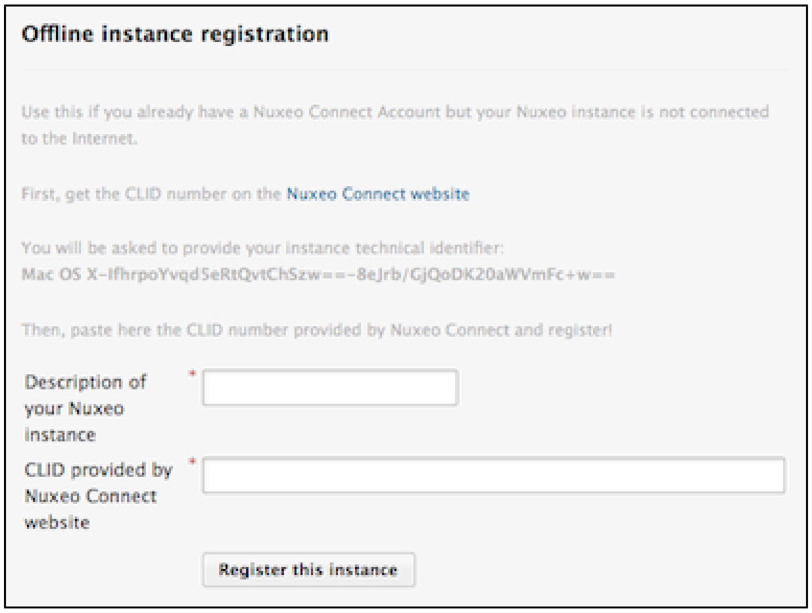
1. Start your Nuxeo instance and connect as an administrator (Administrator/Administrator by default).
2. Click on the **Admin Center** tab.
3. Click on the **Nuxeo Connect** left tab.
4. Authenticate to Nuxeo Connect portal by giving your credentials.
5. Choose which application is concerned among the ones your recorded in Nuxeo Connect.
6. Give a description of your registration, like "Jolene's instance" and indicate if it is a development, qualification or production instance (just for our information).
7. The registration process will end automatically, you can now browse the various tabs of the Update Center area, and set up addons from the Nuxeo Marketplace (see the how to).

Registering Offline

Offline registration can be used when the server doesn't connect to the Internet. It enables to remove the Connect registration footer.

To register your instance for the first time:

1. On the offline server:
   a. Start your Nuxeo instance and connect as an administrator (Administrator/Administrator by default).
   b. Click on the **Admin Center** tab.
   c. Click on the **Nuxeo Connect** left tab.
   d. On the right part of the registration screen, get the **instance technical identifier** called CTID (ex:`Mac OS X-EbMKUsirT9WQszM5mDkaKAp=-BhnJsMDaabDHAQ0A300d6Q==`) and store it in a file so that you can connect to the Internet.



Offline instance registration

Use this if you already have a Nuxeo Connect Account but your Nuxeo instance is not connected to the Internet.

First, get the CLID number on the Nuxeo Connect website

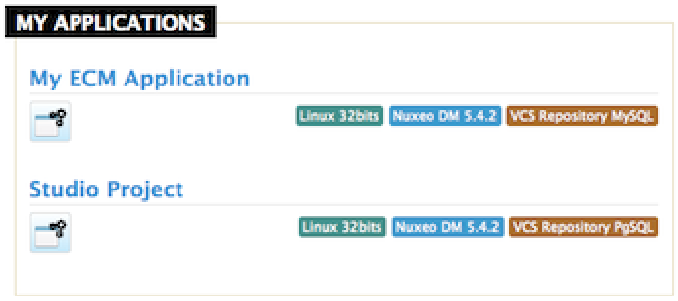
You will be asked to provide your instance technical identifier:
Mac OS X-IfhrpoYvqd5eRtQvtChSzw==-8eJrb/GjQoDK20aWVmFc+w==

Then, paste here the CLID number provided by Nuxeo Connect and register!

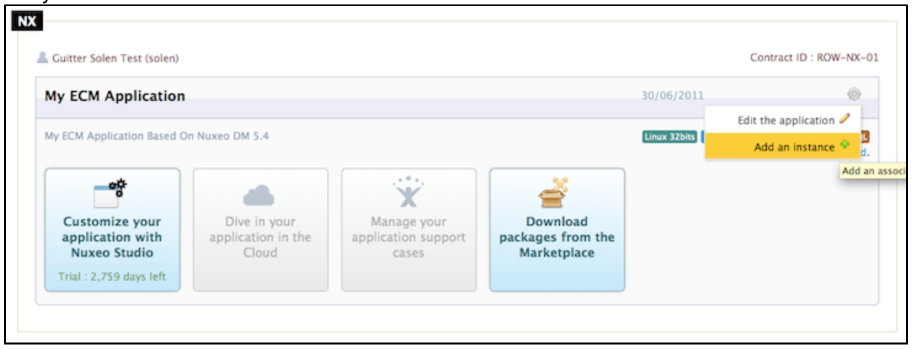
Description of your Nuxeo instance *	
CLID provided by Nuxeo Connect website *	

2. From the Internet-connected-computer:

a. Go to the Nuxeo Connect portal.
b. Click on the Connect application for which you want to register your Nuxeo Platform.

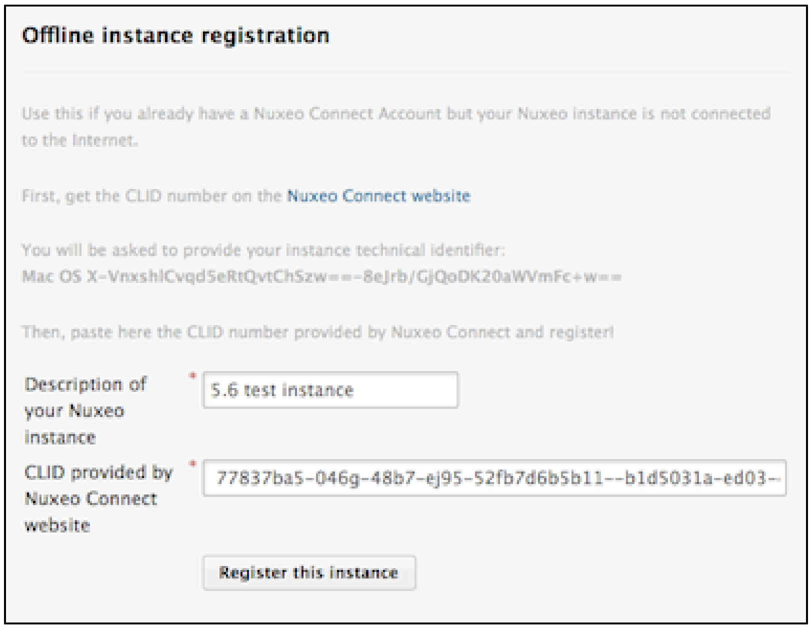


c. Put your mouse over the icon and click **Add a new instance**.


NX
Guitter Solen Test (solen)
Contract ID : ROW-NX-01
My ECM Application
30/06/2011
My ECM Application Based On Nuxeo DM 5.4
Edit the application
Add an instance
Add an associ
Linux 32bits
Customize your application with Nuxeo Studio
Trial : 2,759 days left
Dive in your application in the Cloud
Manage your application support cases
Download packages from the Marketplace

d. Fill in the registration form and submit it.
   The instance is registered. You are given an identifier (CLID) to validate registration from Nuxeo Admin Center.
e. Copy this identifier.

3. On the offline server:
   In the Admin Center, fill in the instance description, paste the CLID from Nuxeo Connect and click on the **Register this instance** butt on.



Offline instance registration

Use this if you already have a Nuxeo Connect Account but your Nuxeo instance is not connected to the Internet.

First, get the CLID number on the Nuxeo Connect website

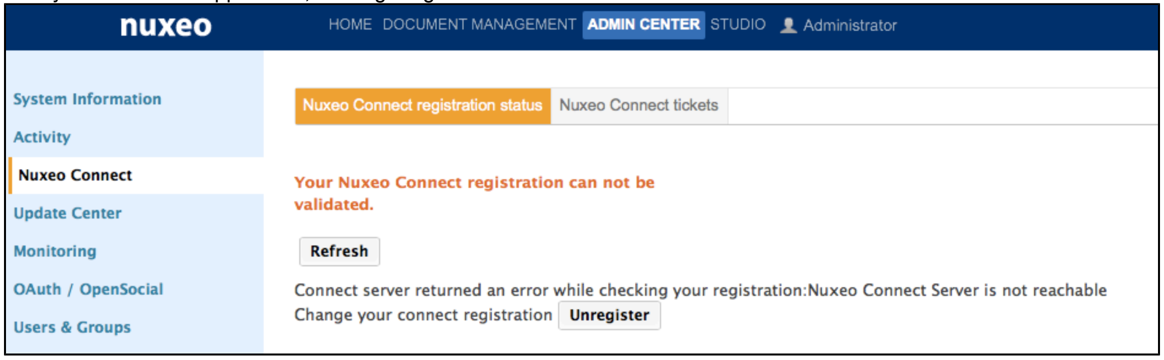
You will be asked to provide your instance technical identifier:
Mac OS X-VnxshlCvqdSeRtQvtChSzw==-8eJrb/GJQoDK20aWVmFc+w==

Then, paste here the CLID number provided by Nuxeo Connect and register!

Description of your Nuxeo instance *	5.6 test instance
CLID provided by Nuxeo Connect website *	77837ba5-046g-48b7-ej95-52fb7d6b5b11--b1d5031a-ed03-

Register this instance

The registration is approved and the registration summary is displayed. The Connect registration footer is not displayed anymore after you browse the application, although registration cannot be validated on the Connect server.



Your Nuxeo Connect registration can not be validated.

Connect server returned an error while checking your registration:Nuxeo Connect Server is not reachable

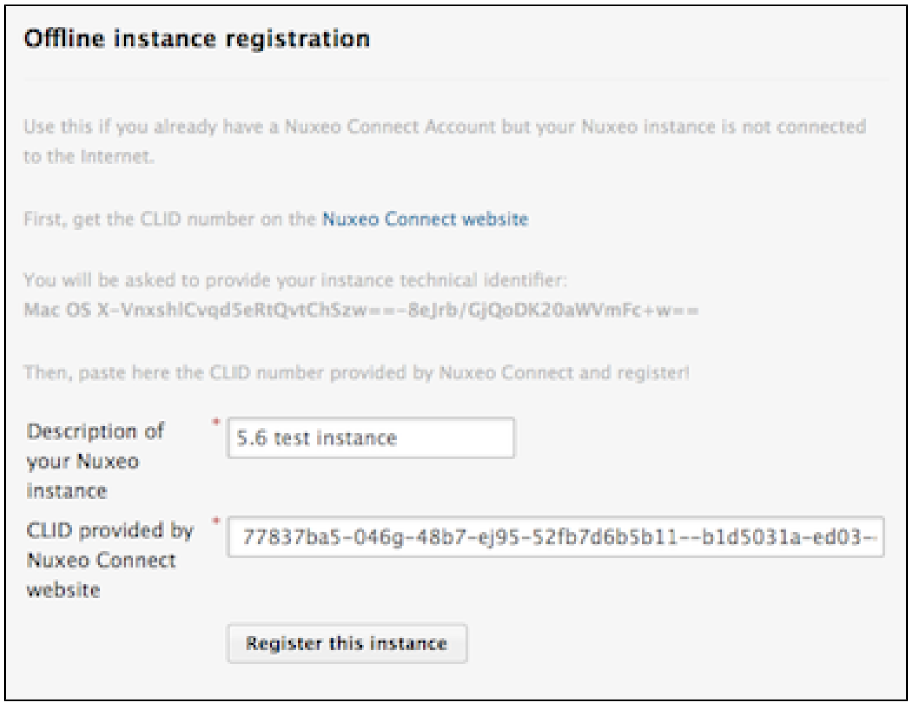
Change your connect registration

Re-Registering your Nuxeo Instance

If you have removed your data from your Nuxeo application, in case of a development instance, for example, you will need to register your instance again.

To re-register your instance:

1. Log in to Nuxeo Connect.
2. In the associated instances, click on your Nuxeo instances link.
   The page listing the associated instances for your project opens.
3. Copy the Identifier of the instance you want to register (Identifier is of the form "fdedaf21-be00-412e-b0ab-f2394479d5f8--885dcf60-5a4a-46e8-a904-0123456789ab").
4. In Nuxeo Admin Center, paste this identifier in the CLID field, fill in the instance description and click on the **Register this instance** b utton.



Offline instance registration

Use this if you already have a Nuxeo Connect Account but your Nuxeo instance is not connected to the Internet.

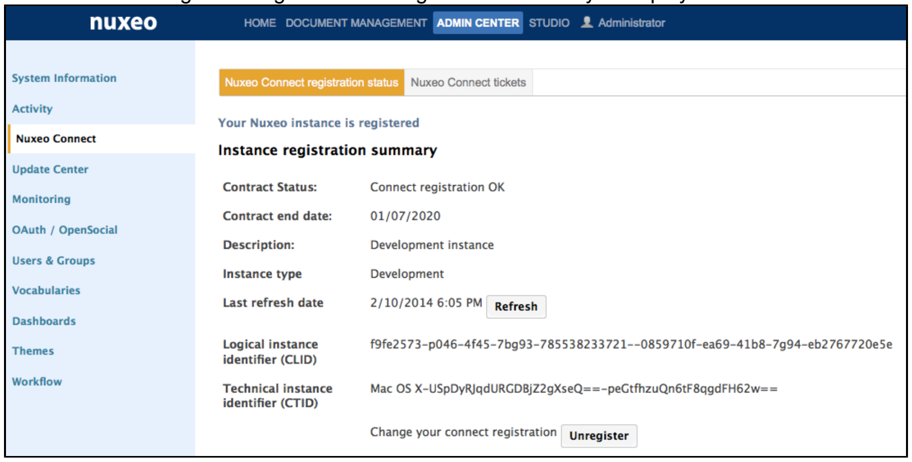
First, get the CLID number on the Nuxeo Connect website

You will be asked to provide your instance technical identifier:
Mac OS X-VnxshlCvqd5eRtQvtChSzw==-8eJrb/GjQoDK20aWVmFc+w==

Then, paste here the CLID number provided by Nuxeo Connect and register!

Description of your Nuxeo instance *	5.6 test instance
CLID provided by Nuxeo Connect website *	77837ba5-046g-48b7-ej95-52fb7d6b5b11--b1d5031a-ed03-

Your instance is registered again and the registration summary is displayed.



Your Nuxeo instance is registered

Instance registration summary

Contract Status:	Connect registration OK
Contract end date:	01/07/2020
Description:	Development instance
Instance type	Development
Last refresh date	2/10/2014 6:05 PM Refresh
Logical instance identifier (CLID)	f9fe2573-p046-4f45-7bg93-785538233721--0859710f-ea69-41b8-7g94-eb2767720e5e
Technical instance identifier (CTID)	Mac OS X-USpDyRJqdURGDBjZ2gXseQ==-peGtfhzuQn6tF8qgdFH62w==
	Change your connect registration Unregister

Installing a New Package on Your Instance

Packages can be installed directly from the Admin Center or from the Marketplace. Packages can be addons bringing new features, hotfixes providing corrections and small improvements. Some Marketplace packages are totally public, not requiring a Connect account to install them. Others can only be installed on instances registered on Nuxeo Connect.

In this section

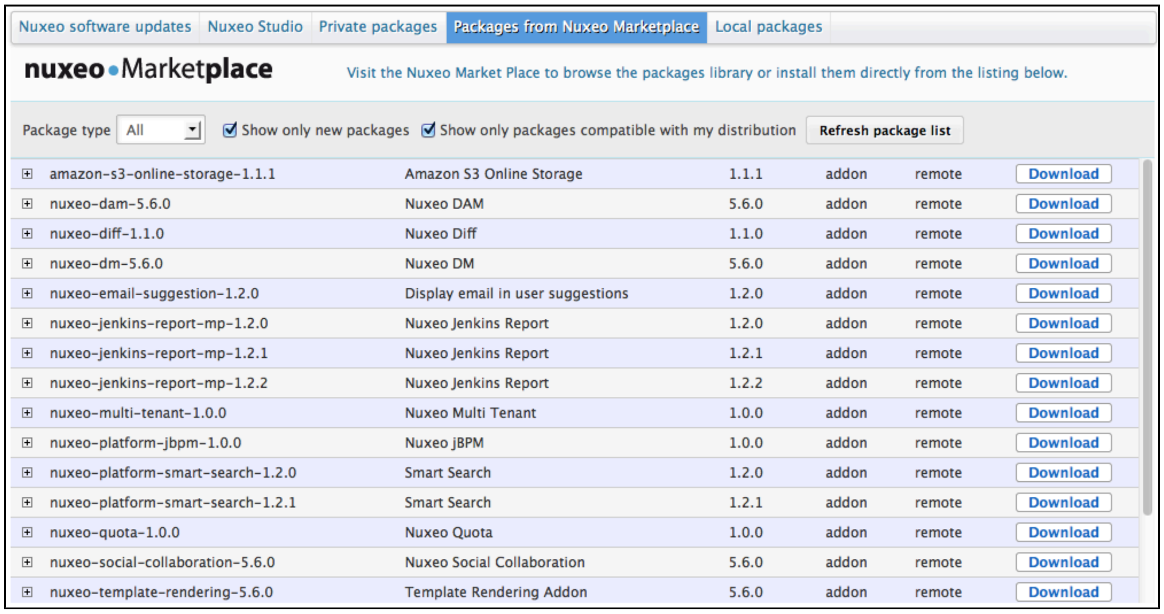
Installing a Package from the Admin Center

The Admin Center includes a section called **Update Center** from which you can easily install hotfixes, updates, addons and your customizations. The Update Center features a **Packages from Marketplace** tab that shows the list of packages available from the Marketplace and allowing you to install these packages directly from your Nuxeo application.

To install a package from the Admin Center:

1. As administrator (Administrator/Administrator by default), in the **Admin Center**, click on the **Update Center** left tab.
2. Click on the **Packages from Nuxeo Marketplace**.
   The list of available packages, including hotfixes and addons, is displayed. By default, only packages compatible with your version of the Nuxeo Platform are listed.

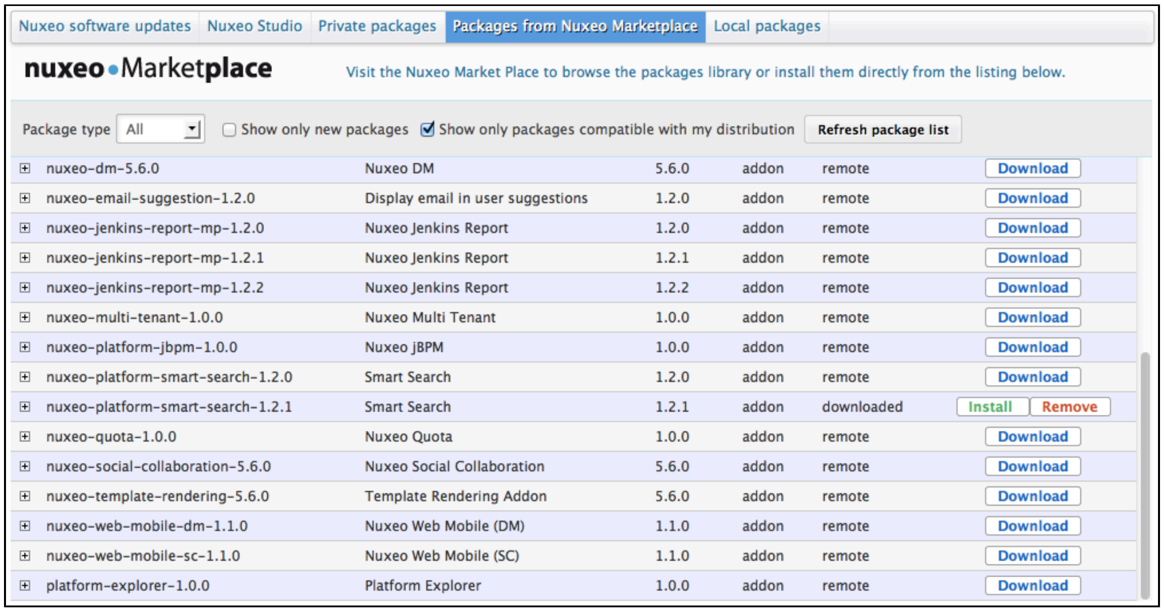
3. Optionally, filter the list of packages:
   - uncheck the **Show only packages compatible with my distribution** box of you want to see all available packages;
   - select a type of package in the drop down list if you want to narrow the list to a package type (addon, hotfix);
   - check the **Show only new packages** box if you want to hide local packages (i.e. installed and downloaded packages) from the list.



Nuxeo software updates
Nuxeo Studio
Private packages
Packages from Nuxeo Marketplace
Local packages
nuxeo•Marketplace
Visit the Nuxeo Market Place to browse the packages library or install them directly from the listing below.
Package type All
☑ Show only new packages
☑ Show only packages compatible with my distribution
Refresh package list

amazon-s3-online-storage-1.1.1	Amazon S3 Online Storage	1.1.1	addon	remote	Download
nuxeo-dam-5.6.0	Nuxeo DAM	5.6.0	addon	remote	Download
nuxeo-diff-1.1.0	Nuxeo Diff	1.1.0	addon	remote	Download
nuxeo-dm-5.6.0	Nuxeo DM	5.6.0	addon	remote	Download
nuxeo-email-suggestion-1.2.0	Display email in user suggestions	1.2.0	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.0	Nuxeo Jenkins Report	1.2.0	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.1	Nuxeo Jenkins Report	1.2.1	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.2	Nuxeo Jenkins Report	1.2.2	addon	remote	Download
nuxeo-multi-tenant-1.0.0	Nuxeo Multi Tenant	1.0.0	addon	remote	Download
nuxeo-platform-jbpm-1.0.0	Nuxeo jBPM	1.0.0	addon	remote	Download
nuxeo-platform-smart-search-1.2.0	Smart Search	1.2.0	addon	remote	Download
nuxeo-platform-smart-search-1.2.1	Smart Search	1.2.1	addon	remote	Download
nuxeo-quota-1.0.0	Nuxeo Quota	1.0.0	addon	remote	Download
nuxeo-social-collaboration-5.6.0	Nuxeo Social Collaboration	5.6.0	addon	remote	Download
nuxeo-template-rendering-5.6.0	Template Rendering Addon	5.6.0	addon	remote	Download

4. Click on the **Download** link of the package you want to install.
   A download in progress page is displayed while the package is being downloaded.
   When download is finished, the list of Marketplace packages is displayed again and the downloaded package has a green **Install** link
   .

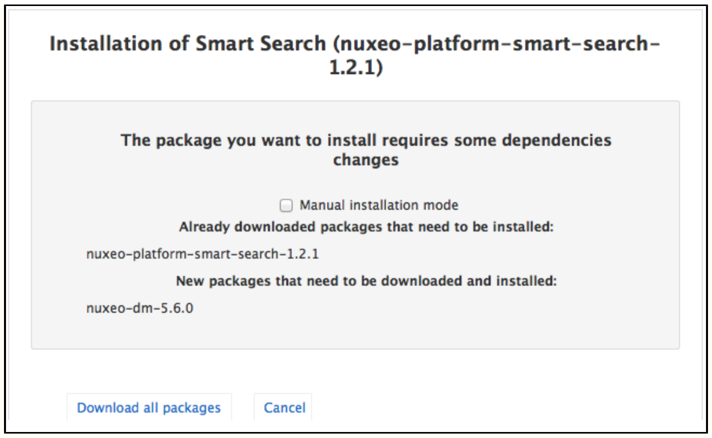


Nuxeo software updates
Nuxeo Studio
Private packages
Packages from Nuxeo Marketplace
Local packages
nuxeo•Marketplace
Visit the Nuxeo Market Place to browse the packages library or install them directly from the listing below.
Package type All
☐ Show only new packages
☑ Show only packages compatible with my distribution
Refresh package list

nuxeo-dm-5.6.0	Nuxeo DM	5.6.0	addon	remote	Download
nuxeo-email-suggestion-1.2.0	Display email in user suggestions	1.2.0	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.0	Nuxeo Jenkins Report	1.2.0	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.1	Nuxeo Jenkins Report	1.2.1	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.2	Nuxeo Jenkins Report	1.2.2	addon	remote	Download
nuxeo-multi-tenant-1.0.0	Nuxeo Multi Tenant	1.0.0	addon	remote	Download
nuxeo-platform-jbpm-1.0.0	Nuxeo jBPM	1.0.0	addon	remote	Download
nuxeo-platform-smart-search-1.2.0	Smart Search	1.2.0	addon	remote	Download
nuxeo-platform-smart-search-1.2.1	Smart Search	1.2.1	addon	downloaded	Install Remove
nuxeo-quota-1.0.0	Nuxeo Quota	1.0.0	addon	remote	Download
nuxeo-social-collaboration-5.6.0	Nuxeo Social Collaboration	5.6.0	addon	remote	Download
nuxeo-template-rendering-5.6.0	Template Rendering Addon	5.6.0	addon	remote	Download
nuxeo-web-mobile-dm-1.1.0	Nuxeo Web Mobile (DM)	1.1.0	addon	remote	Download
nuxeo-web-mobile-sc-1.1.0	Nuxeo Web Mobile (SC)	1.1.0	addon	remote	Download
platform-explorer-1.0.0	Platform Explorer	1.0.0	addon	remote	Download

   The package is also available from the **Local packages** tab of the Update Center.
5. Click on the **Install** link to start the installation.
6. Start the installation by clicking on the **Start** button.

> **Packages with dependencies**
> If the package has some missing dependencies, the **Start** button is not displayed. You are displayed a series of steps to install the required dependencies.
>
> a. If dependency packages are not already in the Local packages, you need to download them. Click on the **Downlo ad all packages** button.


Installation of Smart Search (nuxeo-platform-smart-search-1.2.1)

The package you want to install requires some dependencies changes

☐ Manual installation mode

Already downloaded packages that need to be installed:

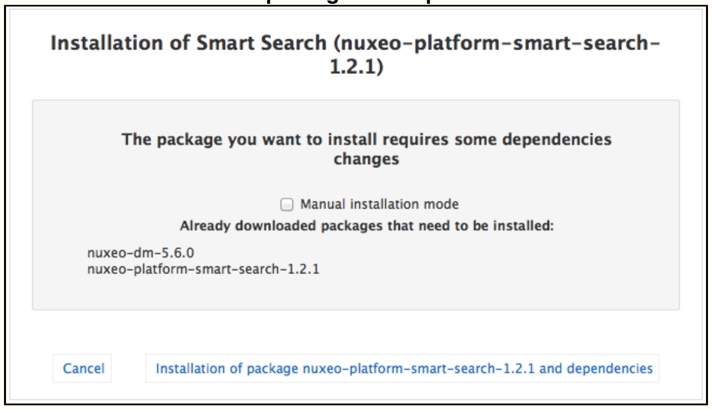
nuxeo-platform-smart-search-1.2.1

New packages that need to be downloaded and installed:

nuxeo-dm-5.6.0

Required packages are downloaded.

b. Click on the **Installation of package and dependencies** button.


Installation of Smart Search (nuxeo-platform-smart-search-1.2.1)

The package you want to install requires some dependencies changes

☐ Manual installation mode

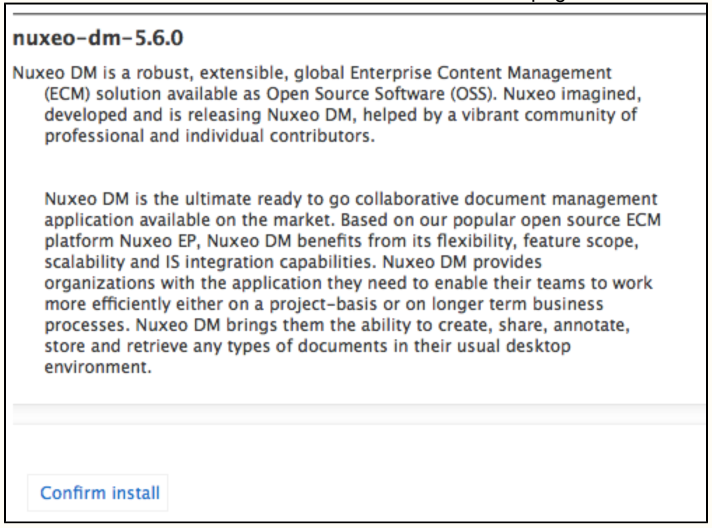
Already downloaded packages that need to be installed:

nuxeo-dm-5.6.0
nuxeo-platform-smart-search-1.2.1

Cancel
Installation of package nuxeo-platform-smart-search-1.2.1 and dependencies

A page detailing the packages to be installed is displayed.

c. Click on the **Confirm install** button at the bottom of the page.

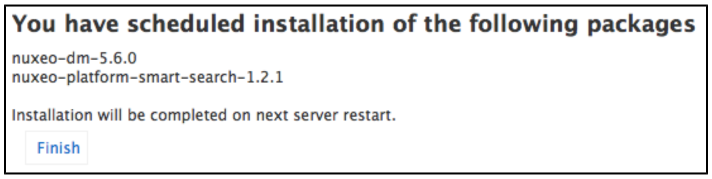

nuxeo-dm-5.6.0

Nuxeo DM is a robust, extensible, global Enterprise Content Management (ECM) solution available as Open Source Software (OSS). Nuxeo imagined, developed and is releasing Nuxeo DM, helped by a vibrant community of professional and individual contributors.

Nuxeo DM is the ultimate ready to go collaborative document management application available on the market. Based on our popular open source ECM platform Nuxeo EP, Nuxeo DM benefits from its flexibility, feature scope, scalability and IS integration capabilities. Nuxeo DM provides organizations with the application they need to enable their teams to work more efficiently either on a project-basis or on longer term business processes. Nuxeo DM brings them the ability to create, share, annotate, store and retrieve any types of documents in their usual desktop environment.

Once the installation is done, a confirmation screen is displayed.

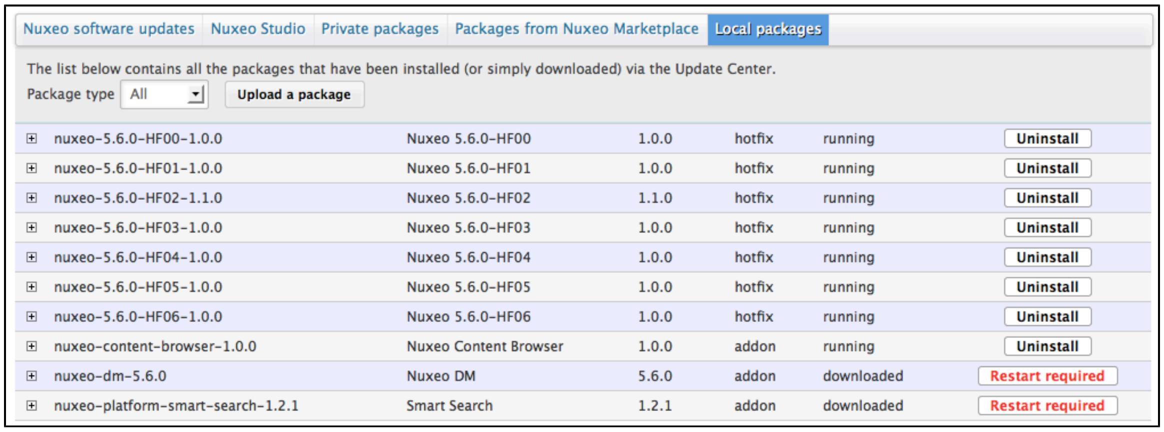
7. Click on the **Finish** button.


You have scheduled installation of the following packages

nuxeo-dm-5.6.0
nuxeo-platform-smart-search-1.2.1

Installation will be completed on next server restart.

The **Install** button is replaced by an **Uninstall** button. The package installation may require to restart the server to complete the installation. In that case, the **Uninstall** button is replaced by a **Restart required** button.

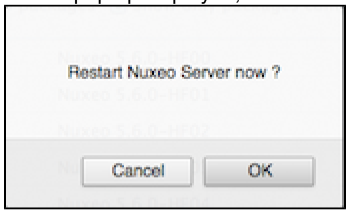

Nuxeo software updates | Nuxeo Studio | Private packages | Packages from Nuxeo Marketplace | Local packages

The list below contains all the packages that have been installed (or simply downloaded) via the Update Center.

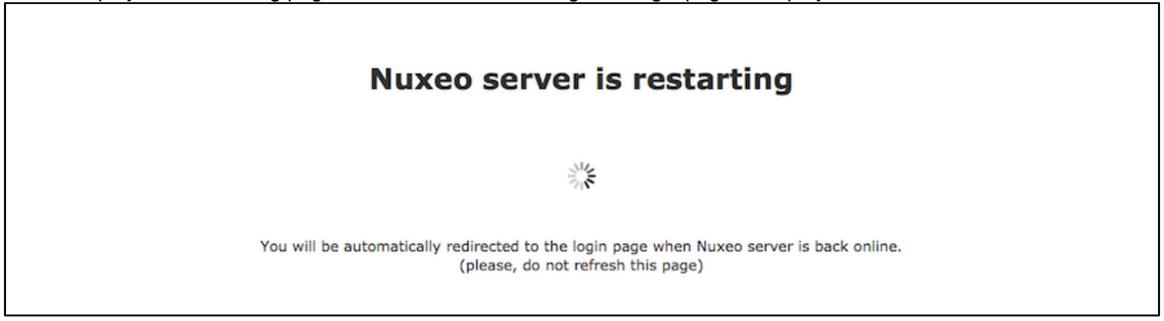
Package type All | Upload a package

nuxeo-5.6.0-HF00-1.0.0	Nuxeo 5.6.0-HF00	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF01-1.0.0	Nuxeo 5.6.0-HF01	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF02-1.1.0	Nuxeo 5.6.0-HF02	1.1.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF03-1.0.0	Nuxeo 5.6.0-HF03	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF04-1.0.0	Nuxeo 5.6.0-HF04	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF05-1.0.0	Nuxeo 5.6.0-HF05	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF06-1.0.0	Nuxeo 5.6.0-HF06	1.0.0	hotfix	running	Uninstall
nuxeo-content-browser-1.0.0	Nuxeo Content Browser	1.0.0	addon	running	Uninstall
nuxeo-dm-5.6.0	Nuxeo DM	5.6.0	addon	downloaded	Restart required
nuxeo-platform-smart-search-1.2.1	Smart Search	1.2.1	addon	downloaded	Restart required

8. If required, click on the **Restart required** button to restart the server.
9. On the pop up displayed, click on the **OK** button to confirm restart.



You're displayed a Restarting page as the server is restarting. The login page is displayed as soon as the server is available again.


Nuxeo server is restarting

You will be automatically redirected to the login page when Nuxeo server is back online.
(please, do not refresh this page)

The server immediately restarts. You're displayed a white page during the server restart. The login page will automatically be displayed when the server is restarted.

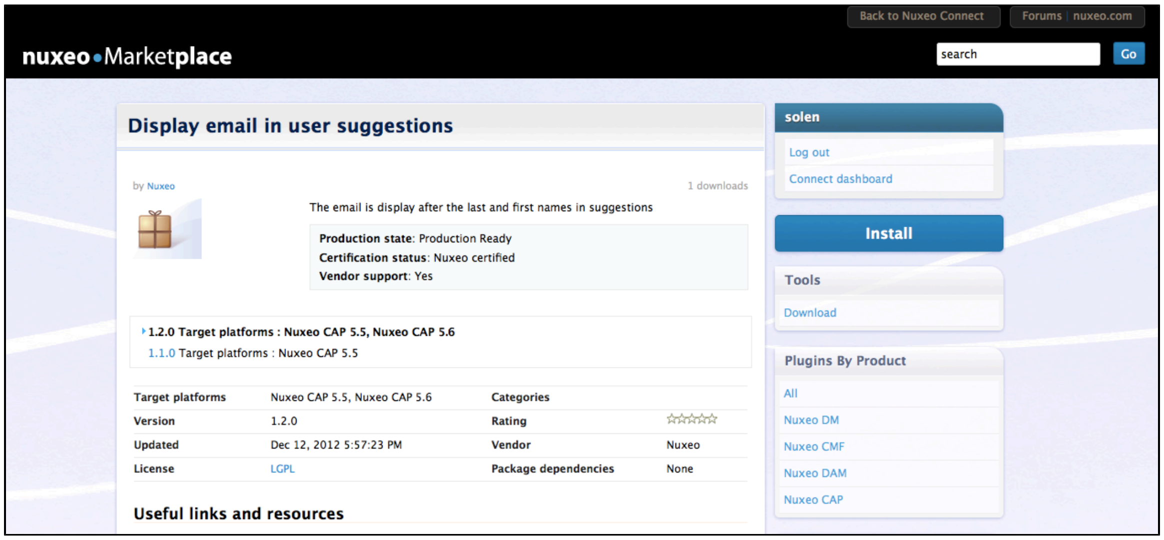
Installing a Package from the Marketplace

There are two ways to install a package from the Marketplace:

1. installing it directly: this requires to be able to access the Nuxeo server as you're on the Marketplace;
2. Downloading it first, and then installing it on the Nuxeo server.

To install the package directly from the Marketplace:

1. On the Marketplace, click on the **Install** button of the package you want to install.



Display email in user suggestions

by Nuxeo 1 downloads

The email is display after the last and first names in suggestions

Production state: Production Ready
Certification status: Nuxeo certified
Vendor support: Yes

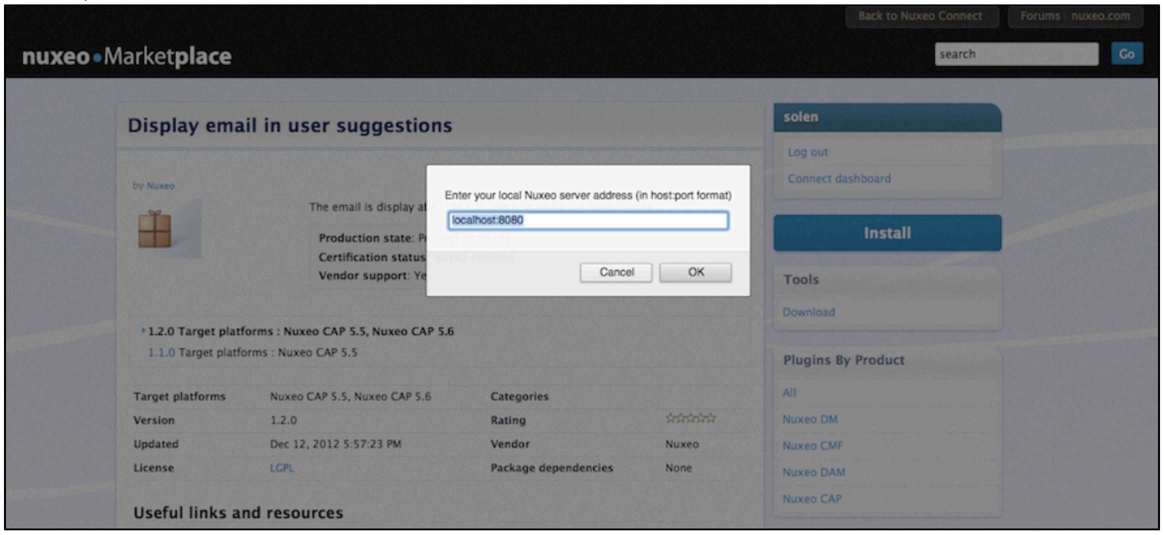
1.2.0 Target platforms : Nuxeo CAP 5.5, Nuxeo CAP 5.6
1.1.0 Target platforms : Nuxeo CAP 5.5

Target platforms	Nuxeo CAP 5.5, Nuxeo CAP 5.6	Categories	
Version	1.2.0	Rating	☆☆☆☆☆
Updated	Dec 12, 2012 5:57:23 PM	Vendor	Nuxeo
License	LGPL	Package dependencies	None

Useful links and resources

A window pops up.

2. Type the URL of the Nuxeo server on which you want to set up the package (and on which you have administrator credentials). For instance, `localhost:8080`.

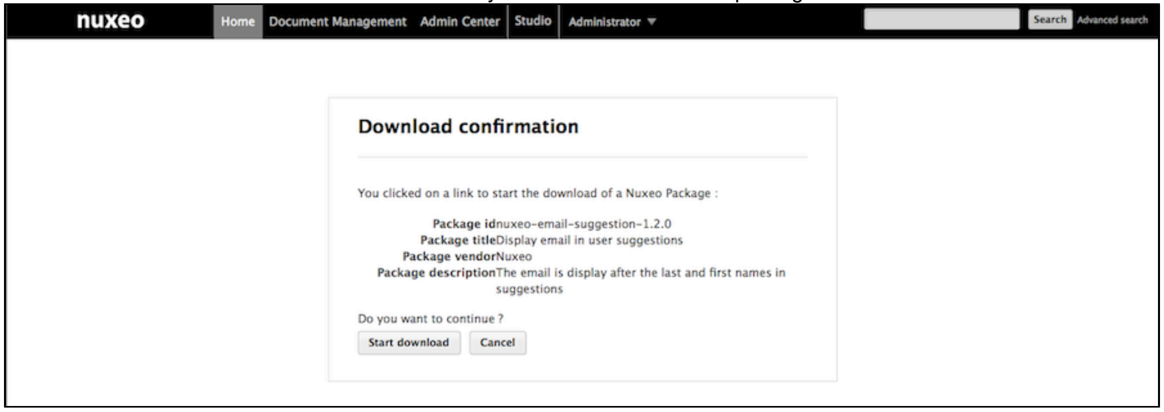


Enter your local Nuxeo server address (in host:port format)

localhost:8080

Your Nuxeo application opens in a new window.

3. Log in as an administrator, if you are not already.
A confirmation page is displayed.
4. Click on the **Start download** button to confirm that you want to download the package.



Download confirmation

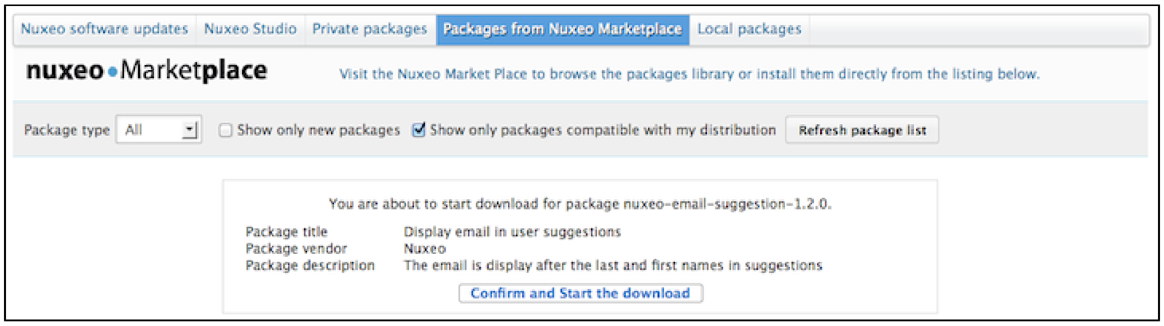
You clicked on a link to start the download of a Nuxeo Package :

Package idnuxeo-email-suggestion-1.2.0
Package title Display email in user suggestions
Package vendor Nuxeo
Package description The email is display after the last and first names in suggestions

Do you want to continue ?

You are directed in the Admin Center.

5. Click on the **Confirm and Start download** button to confirm downloading.



Nuxeo software updates | Nuxeo Studio | Private packages | Packages from Nuxeo Marketplace | Local packages

nuxeo•Marketplace — Visit the Nuxeo Market Place to browse the packages library or install them directly from the listing below.

Package type All | Show only new packages | ✓ Show only packages compatible with my distribution | Refresh package list

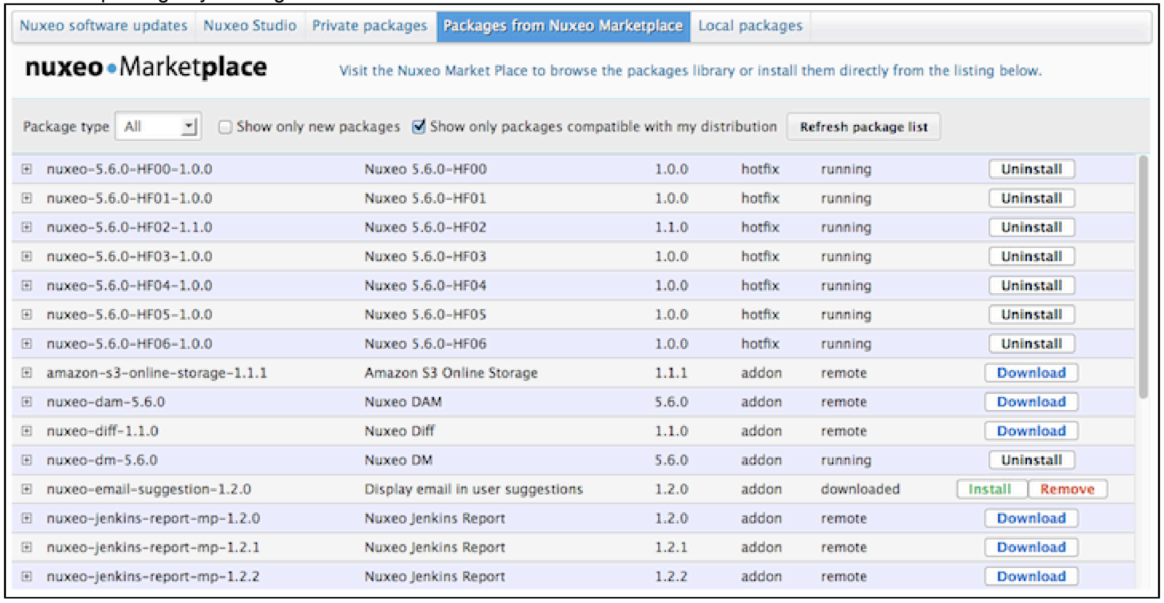
You are about to start download for package nuxeo-email-suggestion-1.2.0.

Package title	Display email in user suggestions
Package vendor	Nuxeo
Package description	The email is display after the last and first names in suggestions

Confirm and Start the download

Once the package has been downloaded, you are displayed the **Packages from Nuxeo Marketplace** tab of the Update Center. The package has a green **Install** button and is also available from the **Local packages** tab of the Update Center.

6. Install the package by clicking on the **Install** link.



Nuxeo software updates | Nuxeo Studio | Private packages | Packages from Nuxeo Marketplace | Local packages

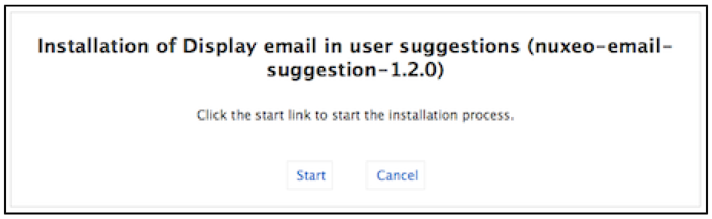
nuxeo•Marketplace — Visit the Nuxeo Market Place to browse the packages library or install them directly from the listing below.

Package type All | Show only new packages | ✓ Show only packages compatible with my distribution | Refresh package list

nuxeo-5.6.0-HF00-1.0.0	Nuxeo 5.6.0-HF00	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF01-1.0.0	Nuxeo 5.6.0-HF01	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF02-1.1.0	Nuxeo 5.6.0-HF02	1.1.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF03-1.0.0	Nuxeo 5.6.0-HF03	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF04-1.0.0	Nuxeo 5.6.0-HF04	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF05-1.0.0	Nuxeo 5.6.0-HF05	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF06-1.0.0	Nuxeo 5.6.0-HF06	1.0.0	hotfix	running	Uninstall
amazon-s3-online-storage-1.1.1	Amazon S3 Online Storage	1.1.1	addon	remote	Download
nuxeo-dam-5.6.0	Nuxeo DAM	5.6.0	addon	remote	Download
nuxeo-diff-1.1.0	Nuxeo Diff	1.1.0	addon	remote	Download
nuxeo-dm-5.6.0	Nuxeo DM	5.6.0	addon	running	Uninstall
nuxeo-email-suggestion-1.2.0	Display email in user suggestions	1.2.0	addon	downloaded	Install Remove
nuxeo-jenkins-report-mp-1.2.0	Nuxeo Jenkins Report	1.2.0	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.1	Nuxeo Jenkins Report	1.2.1	addon	remote	Download
nuxeo-jenkins-report-mp-1.2.2	Nuxeo Jenkins Report	1.2.2	addon	remote	Download

A confirmation page is displayed.

7. Click on **Start** to confirm installation.



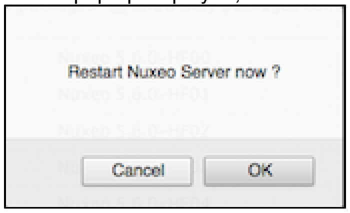
Installation of Display email in user suggestions (nuxeo-email-suggestion-1.2.0)

Click the start link to start the installation process.

Start | Cancel

Once the set up is achieved, a message from the server confirms that the installation was performed correctly.

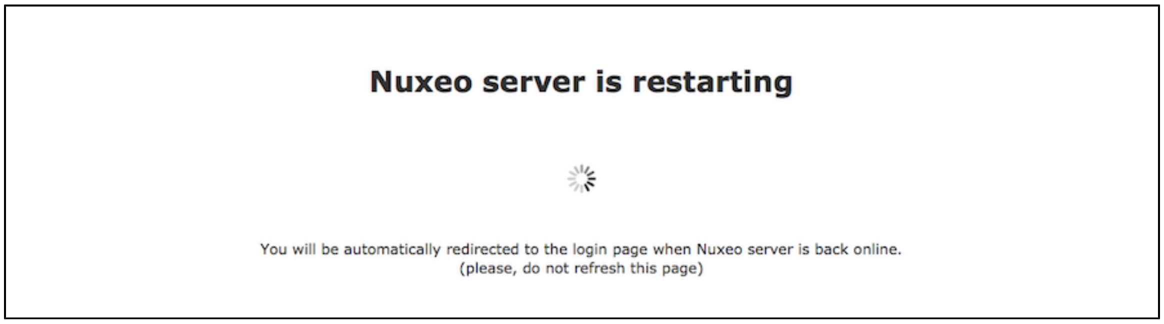
8. If required, restart the server by clicking the **Restart required** button that replaces **Install** and **Remove** buttons. Otherwise, installation is completed and you're displayed an **Uninstall** button.
9. On the pop up displayed, click on the **OK** button to confirm restart.



Restart Nuxeo Server now ?

Cancel | OK

You're displayed a Restarting page as the server is restarting. The login page is displayed as soon as the server is available again.



Nuxeo server is restarting

You will be automatically redirected to the login page when Nuxeo server is back online.
(please, do not refresh this page)

Offline Installation

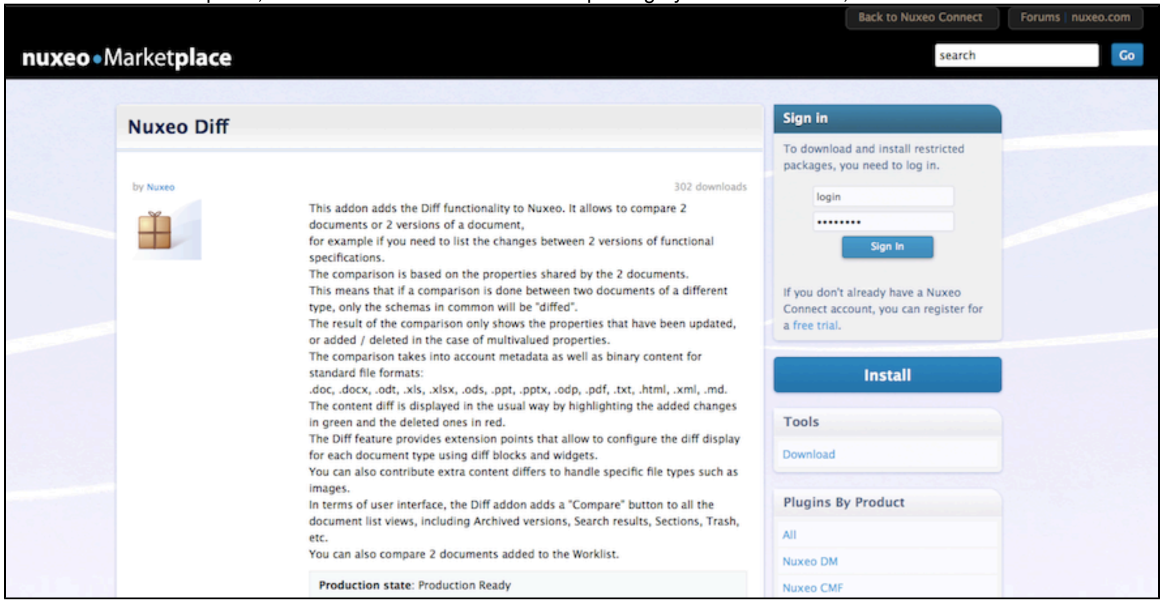
It is possible to install packages available on the Nuxeo Marketplace even if your server is not connected to the Internet. This takes two steps:

1. Download the package from the Marketplace.
2. Upload the package from the Update Center.

To download the package you want to install from the Marketplace:

Depending on the package you want to install, you may need to be logged in to the Marketplace to download the package.

1. On the Nuxeo Marketplace, click on the **Download** link of the package you want to install, in the **Tools** box.



Nuxeo Diff

by Nuxeo

302 downloads

This addon adds the Diff functionality to Nuxeo. It allows to compare 2 documents or 2 versions of a document,
for example if you need to list the changes between 2 versions of functional specifications.
The comparison is based on the properties shared by the 2 documents.
This means that if a comparison is done between two documents of a different type, only the schemas in common will be "diffed".
The result of the comparison only shows the properties that have been updated, or added / deleted in the case of multivalued properties.
The comparison takes into account metadata as well as binary content for standard file formats:
.doc, .docx, .odt, .xls, .xlsx, .ods, .ppt, .pptx, .odp, .pdf, .txt, .html, .xml, .md.
The content diff is displayed in the usual way by highlighting the added changes in green and the deleted ones in red.
The Diff feature provides extension points that allow to configure the diff display for each document type using diff blocks and widgets.
You can also contribute extra content differs to handle specific file types such as images.
In terms of user interface, the Diff addon adds a "Compare" button to all the document list views, including Archived versions, Search results, Sections, Trash, etc.
You can also compare 2 documents added to the Worklist.

Production state: Production Ready

Sign in

To download and install restricted packages, you need to log in.

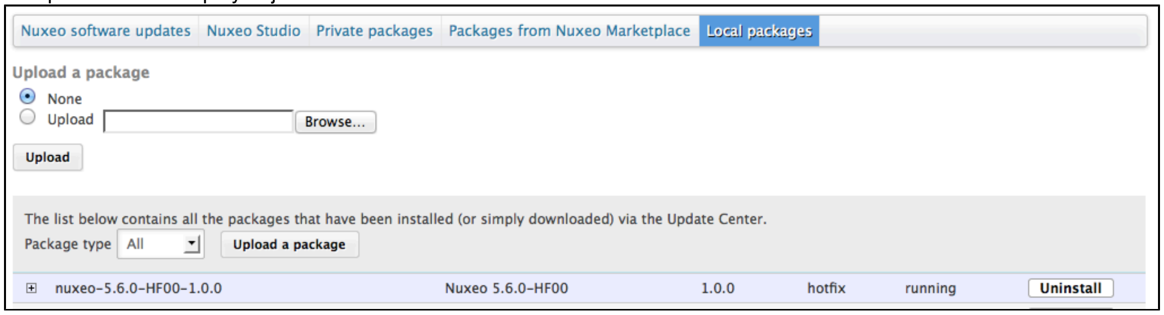
login
••••••••
Sign In

If you don't already have a Nuxeo Connect account, you can register for a free trial.

2. Save the .zip file on a disc that is accessible by the Nuxeo server or directly on a storage device.

To install the downloaded package:

1. As administrator (Administrator/Administrator by default), in the **Admin Center**, click on the **Update Center** left tab.
2. Click on the **Local packages** tab.
3. Click on the **Upload a package** button.
   An upload form is displayed just below the tabs.



Upload a package

[✓ None]
Upload
Browse...
Upload

The list below contains all the packages that have been installed (or simply downloaded) via the Update Center.

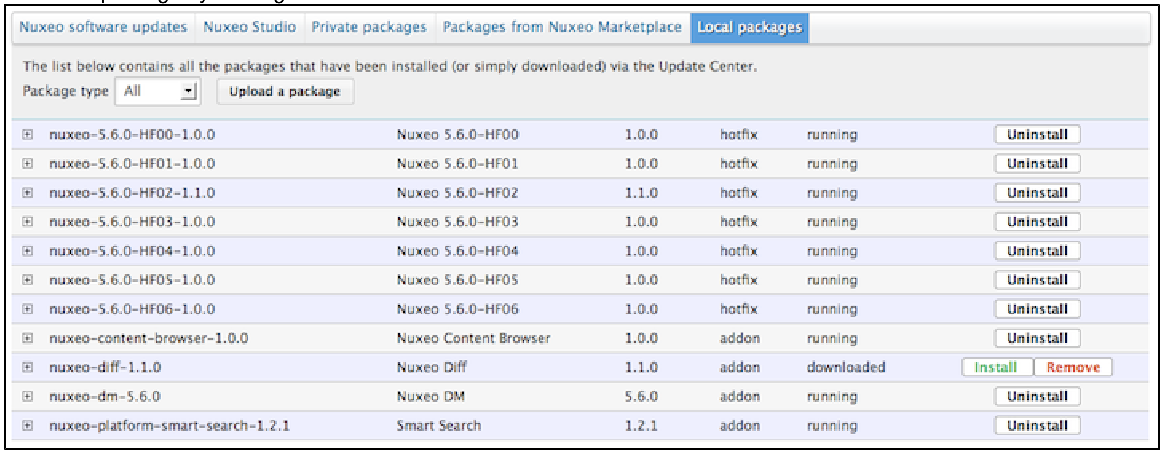
Package type: All
Upload a package

nuxeo-5.6.0-HF00-1.0.0	Nuxeo 5.6.0-HF00	1.0.0	hotfix	running	Uninstall

4. Click on the **Browse** button to select the package .zip package file.
5. Click on the **Upload** button.

The package is uploaded to the server and ready to be installed.

6. Install the package by clicking on the **Install** link.



Nuxeo software updates | Nuxeo Studio | Private packages | Packages from Nuxeo Marketplace | Local packages

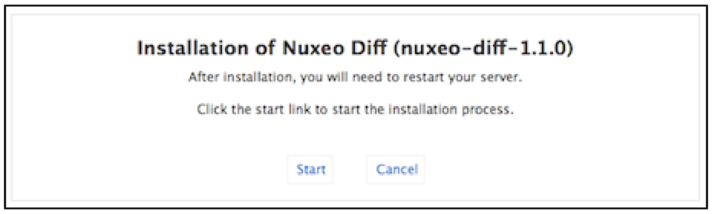
The list below contains all the packages that have been installed (or simply downloaded) via the Update Center.

Package type All | Upload a package

nuxeo-5.6.0-HF00-1.0.0	Nuxeo 5.6.0-HF00	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF01-1.0.0	Nuxeo 5.6.0-HF01	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF02-1.1.0	Nuxeo 5.6.0-HF02	1.1.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF03-1.0.0	Nuxeo 5.6.0-HF03	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF04-1.0.0	Nuxeo 5.6.0-HF04	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF05-1.0.0	Nuxeo 5.6.0-HF05	1.0.0	hotfix	running	Uninstall
nuxeo-5.6.0-HF06-1.0.0	Nuxeo 5.6.0-HF06	1.0.0	hotfix	running	Uninstall
nuxeo-content-browser-1.0.0	Nuxeo Content Browser	1.0.0	addon	running	Uninstall
nuxeo-diff-1.1.0	Nuxeo Diff	1.1.0	addon	downloaded	Install Remove
nuxeo-dm-5.6.0	Nuxeo DM	5.6.0	addon	running	Uninstall
nuxeo-platform-smart-search-1.2.1	Smart Search	1.2.1	addon	running	Uninstall

A confirmation page is displayed.

7. Click on **Start** to confirm installation.



Installation of Nuxeo Diff (nuxeo-diff-1.1.0)

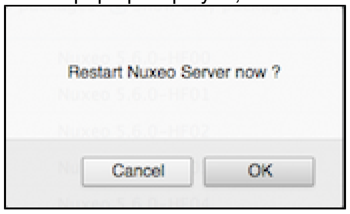
After installation, you will need to restart your server.

Click the start link to start the installation process.

Start Cancel

Once the set up is achieved, a message from the server confirms that the installation was performed correctly.

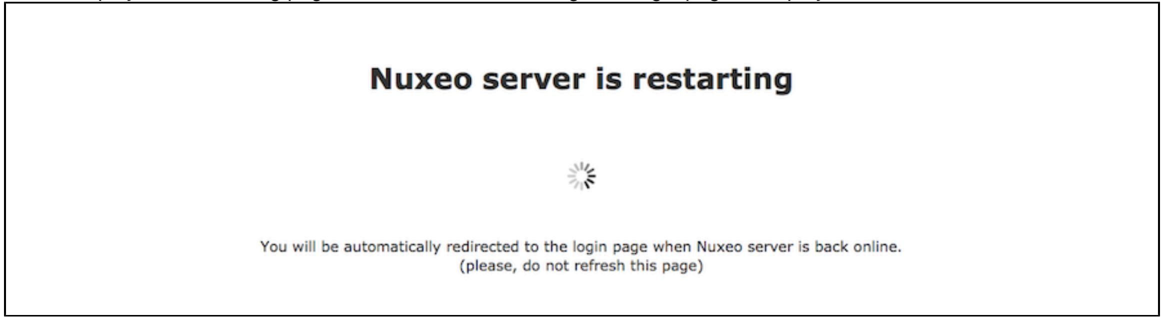
8. If required, restart the server by clicking the **Restart required** button that replaces **Install** and **Remove** buttons. Otherwise, installation is completed and you're displayed an **Uninstall** button.
9. On the pop up displayed, click on the **OK** button to confirm restart.



Restart Nuxeo Server now ?

Cancel OK

You're displayed a Restarting page as the server is restarting. The login page is displayed as soon as the server is available again.



Nuxeo server is restarting

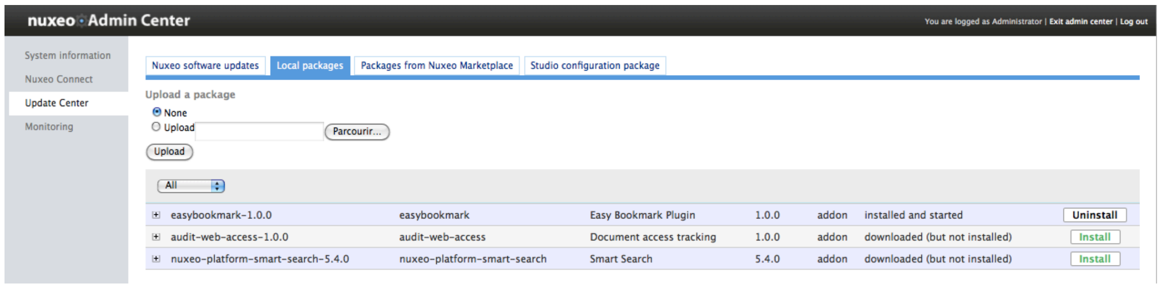
You will be automatically redirected to the login page when Nuxeo server is back online.
(please, do not refresh this page)

Uninstalling a Package

Uninstalling a package can be done from the Admin Center only.

To uninstall a package:

1. In the Admin Center, go on the **Local packages** tab of the **Update Center**.
   The list of packages that you have downloaded and possibly installed is displayed.


nuxeo Admin Center — You are logged as Administrator | Exit admin center | Log out

System information · Nuxeo Connect · Update Center · Monitoring

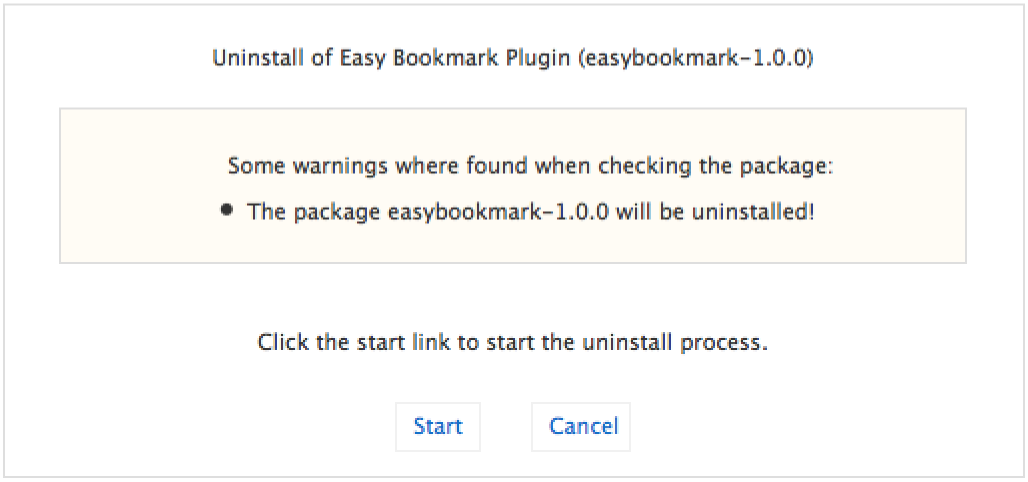
Nuxeo software updates | Local packages | Packages from Nuxeo Marketplace | Studio configuration package

Upload a package
(●) None
( ) Upload [ ] Parcourir...
Upload

All

easybookmark-1.0.0	easybookmark	Easy Bookmark Plugin	1.0.0	addon	installed and started	Uninstall
audit-web-access-1.0.0	audit-web-access	Document access tracking	1.0.0	addon	downloaded (but not installed)	Install
nuxeo-platform-smart-search-5.4.0	nuxeo-platform-smart-search	Smart Search	5.4.0	addon	downloaded (but not installed)	Install

2. Click on the **Uninstall** link of the package you want to uninstall from your application.
   A confirmation message is displayed.
3. Click on the **Start** button to confirm you want to uninstall the package.


Uninstall of Easy Bookmark Plugin (easybookmark-1.0.0)

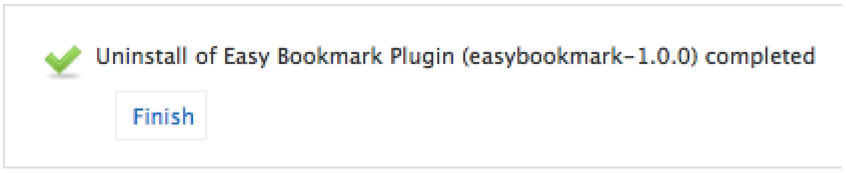
Some warnings where found when checking the package:

- The package easybookmark-1.0.0 will be uninstalled!

Click the start link to start the uninstall process.

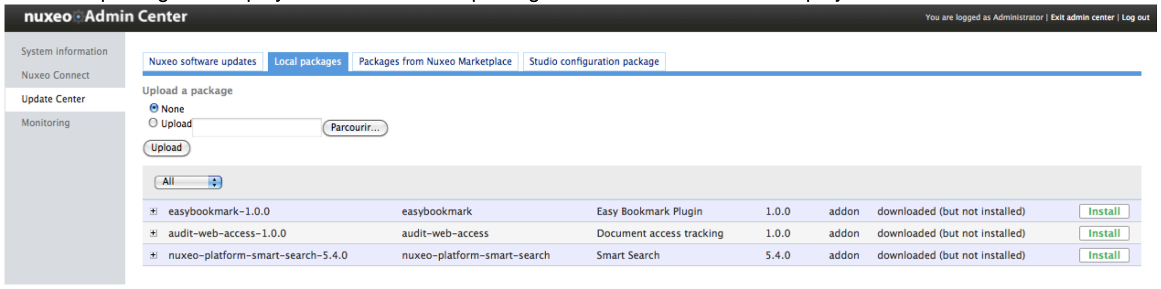
Start Cancel

4. When uninstallation is done, click on the **Finish** button.


Uninstall of Easy Bookmark Plugin (easybookmark-1.0.0) completed

Finish

The list of packages is displayed. The uninstalled package now has an **Install** link displayed.


nuxeo Admin Center — You are logged as Administrator | Exit admin center | Log out

System information · Nuxeo Connect · Update Center · Monitoring

Nuxeo software updates | Local packages | Packages from Nuxeo Marketplace | Studio configuration package

Upload a package
(●) None
( ) Upload [ ] Parcourir...
Upload

All

easybookmark-1.0.0	easybookmark	Easy Bookmark Plugin	1.0.0	addon	downloaded (but not installed)	Install
audit-web-access-1.0.0	audit-web-access	Document access tracking	1.0.0	addon	downloaded (but not installed)	Install
nuxeo-platform-smart-search-5.4.0	nuxeo-platform-smart-search	Smart Search	5.4.0	addon	downloaded (but not installed)	Install

If the package doesn't support hotreload, you need to reboot the server so the unistallation is completed.

Updating your Instance with Studio Configuration

Just like you can install Nuxeo Marketplace packages on your Nuxeo instance from the Admin Center, you can also update it with your customization done in Nuxeo Studio. The **Studio configuration package** tab lists all the tagged versions of your Studio projects.

But it also enables you to load your current Nuxeo Studio configuration. Hot reload of your Nuxeo Studio configuration enables you to test directly your changes in your Nuxeo application, without having to restart the server or logout.

To apply your Studio changes to your application:

1. Make sure you have registered your instance.
2. In your application, go on **Nuxeo Admin Center** > **Update Center** > **Nuxeo Studio**.

3. Click on the button **Update**.
   The configuration is reloaded: if you configured a new button for instance, bound to a content automation chain, you can just test it directly.

Marketplace Add-Ons

The Nuxeo Marketplace is Nuxeo's ECM application store. It offers plug-ins and packages that enable you to easily add features to your Nuxeo application. Packages are listed by module (Nuxeo DM, Nuxeo DAM), and by categories (workflow, collaborative tools...). The list of available packages is available to everyone, but some packages require a Nuxeo Connect account to be able to install packages.

The Marketplace offers packages aimed at developers and other that provide new features to end-users. Most of the packages can be installe d from the Update Center very easily and don't require any additional installation or configuration step. However, some other add-ons, typically connectors with other systems, involve some additional configuration.

Each package has a dedicated page on the Marketplace, that describes the feature the package enables, if there are prerequisites, etc. Here is the information available about the packages from the Marketplace:

- **Production state**: Indicates if the package is approved by Nuxeo or is still in testing phase.
- **Certification status**: Indicates if the packages has been certified by Nuxeo or not.
- **Vendor support**: Indicates if the package is covered by Nuxeo Connect support contracts.
- **Type**: Possible types are: add-ons will provide new features, hot-fixes provide corrections, and Studio packages install your Studio customizations in your instance.
- **Last version**: Most recent version number of the plug-in.
- **Updated**: Date on which the package was last updated.
- **Target platforms**: Nuxeo applications on which you can install the package.
- **License**: License applied to the package.
- **Categories**: List of categories the package belongs to.
- **Rating**: Comments on the package.
- **Vendor**: Name of the person or company who developed the package.
- **Package dependencies**: indicates if there are some requirements for the package to be correctly installed.
- **Hot-reload support**: Indicates if the plug-in is immediately functional (i.e. no server reboot required).

Although most packages are installed in a few clicks from the Update Center, some of them require specific installation or configuration steps. Below is the list of available Marketplace packages and their documentation.

The page Marketplace addons does not exist.

Amazon S3 Online Storage

The Amazon S3 Online Storage is a Nuxeo Binary Manager for S3. It stores Nuxeo's binaries (the attached documents) in an Amazon S3 bucket.

Before You Start

You should be familiar with Amazon S3 and be in possession of your credentials.

Installing the Package

Use the Update Center to install the package from the Nuxeo Marketplace.

In this section:

Configuring the Package

In order to configure the package, you will need to change a few Nuxeo templates, and provide values for the configuration variables that define your S3 credentials, bucket and encryption choices

Specifying Your Amazon S3 Parameters

In `nuxeo.conf`, add the following lines:

```
nuxeo.s3storage.bucket=your_s3_bucket_name
nuxeo.s3storage.awsid=your_AWS_ACCESS_KEY_ID
nuxeo.s3storage.awssecret=your_AWS_SECRET_ACCESS_KEY
```

If you installed the bundle JAR manually instead of using the marketplace package you will also need:

```
nuxeo.core.binarymanager=org.nuxeo.ecm.core.storage.sql.S3BinaryManager
```

The bucket name is unique across all of Amazon, you should find something original and specific.

The file `nuxeo.conf` now contains S3 secret access keys, you should protect it from prying eyes.

You can also add the following optional parameters:

```
nuxeo.s3storage.region=us-west-1
nuxeo.s3storage.cachesize=100MB
```

The region code can be:

- for us-east-1 (the default), don't specify this parameter
- for us-west-1 (Northern California), use `us-west-1`
- for eu-west-1 (Ireland), use `EU`
- for ap-southeast-1 (Singapore), use `ap-southeast-1`

Since 5.6, you can also use:

- for us-west-2 (Oregon), `us-west-2`
- for ap-southeast-2 (Tokyo), use `ap-southeast-2`
- for sa-east-1 (Sao Paulo), use `sa-east-1`

Crypto Options

With S3 you have the option of storing your data encrypted (note that the local cache will *not* be encrypted).

The S3 Binary Manager can use a keystore containing a keypair, but there are a few caveats to be aware of:

- The Sun/Oracle JDK doesn't always allow the AES256 cipher which the AWS SDK uses internally. Depending on the US export restrictions for your country, you may be able to modify your JDK to use AES256 by installing the "Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files". See the following link to download the files and installation instructions: http://www.oracle.com/technetwork/java/javase/downloads/index.html

- Don't forget to specify the key algorithm if you create your keypair with the `keytool` command, as this won't work with the default (DSA). The S3 Binary Manager has been tested with a keystore generated with this command:

  ```
  keytool -genkeypair -keystore </path/to/keystore/file> -alias <key alias>
  -storepass <keystore password> -keypass <key password> -dname <key
  distinguished name> -keyalg RSA
  ```

  If you get `keytool error: java.io.IOException: Incorrect AVA format`, then ensure that the distinguished name parameter has a form such as: `-dname "CN=AWS S3 Key , O=example, DC=com"`.

Don't forget to **make backups of the `/path/to/keystore/file` file** along with the **store password, key alias and key password** . If you lose them (for instance if the EC2 machine hosting the Nuxeo instance with the original keystore is lost) you will lose the ability to recover any encrypted blob from the S3 backet.

With all that above in mind, here are the crypto options that you can add to `nuxeo.conf` (they are all mandatory once you specify a keystore):

```
nuxeo.s3storage.crypt.keystore.file=/absolute/path/to/the/keystore/file
nuxeo.s3storage.crypt.keystore.password=the_keystore_password
nuxeo.s3storage.crypt.key.alias=the_key_alias
nuxeo.s3storage.crypt.key.password=the_key_password
```

The Nuxeo `S3BinaryManager` class is using S3 Client-Side Encryption instead of S3 Server-Side Encrytption. CSE is safer than SSE. With CSE an attacker need both access to the **AWS credentials and the key** to be able to access the unencrypted data while SSE will only require the potential attacker to provide the **AWS credentials**.

Connection Pool Options

Since Nuxeo 5.8.0-HF06 (and Nuxeo 5.9.2 and Nuxeo 5.6.0-HF30) you can configure the internal S3 connection pool. This pool has a size of 50 by default, so if you've configured Nuxeo to use more sessions than this and all the sessions are accessing S3, you may run out of connections.

The following parameters can be used to change some connection pool parameters (the defaults are shown):

```
nuxeo.s3storage.connection.max=50
nuxeo.s3storage.connection.retry=3
nuxeo.s3storage.connection.timeout=50000
nuxeo.s3storage.socket.timeout=50000
```

The timeouts are expressed in milliseconds.

You can read more about these parameters on the AWS ClientConfiguration documentation page.

Checking Your Configuration

To check that installation went well, you can check your startup logs and look for a line like:

```
INFO  [S3BinaryManager] Repository 'default' using S3BinaryManager
```

Don't forget to enable the `INFO` level for the group `org.nuxeo` in `$NUXEO_HOME/lib/log4j.xml` to see INFO level messages from Nuxeo classes.

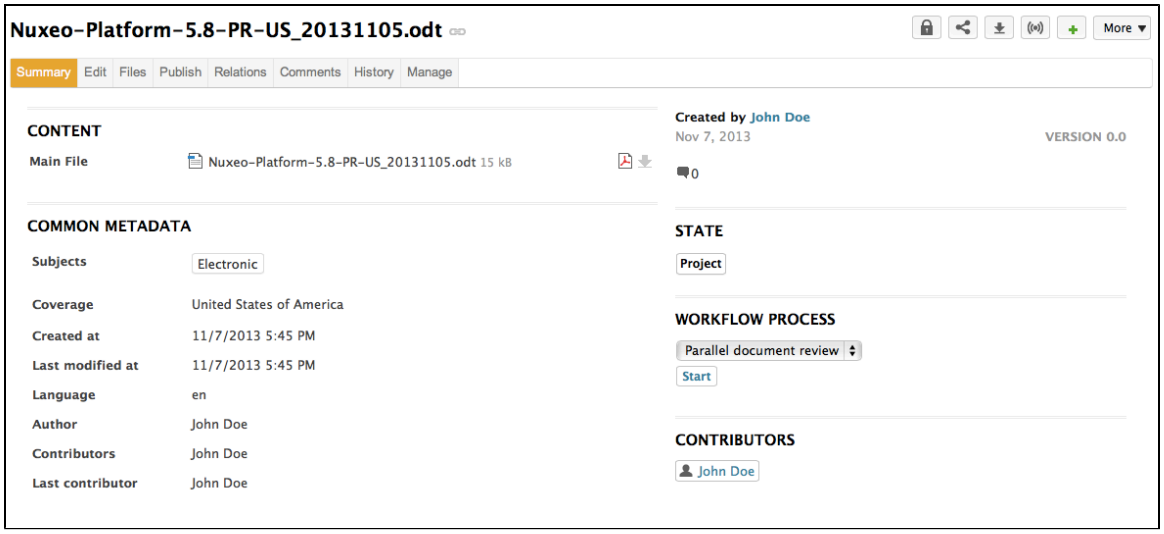
If your configuration is incorrect, this line will be followed by some error messages describing the problems encountered.

Automated Document Categorization

The Automated Document Categorization package enables the system to automatically fill in some metadata of the document when it is created, from the document's content.

The Automated Document Categorization package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, new documents directly have some metadata filled in.


Nuxeo-Platform-5.8-PR-US_20131105.odt

Summary | Edit | Files | Publish | Relations | Comments | History | Manage

CONTENT

Main File	Nuxeo-Platform-5.8-PR-US_20131105.odt 15 kB

COMMON METADATA

Subjects	Electronic
Coverage	United States of America
Created at	11/7/2013 5:45 PM
Last modified at	11/7/2013 5:45 PM
Language	en
Author	John Doe
Contributors	John Doe
Last contributor	John Doe

Created by John Doe
Nov 7, 2013
VERSION 0.0
0

STATE
Project

WORKFLOW PROCESS
Parallel document review
Start

CONTRIBUTORS
John Doe

Other documentation about this package

Automated Document Categorization (Nuxeo Platform User Documentation - 5.8)

Bulk Document Importer

Available for any Nuxeo platform-based application, the Bulk document importer package enables mass document import in a Nuxeo repository. A single HTTP query launches a full, multi-threaded import from the server file system.

Usage

The file importer comes as a Java library (with the Nuxeo Runtime Service) and a sample JAX-RS interface to launch, monitor and abort import jobs.

Quick Start

To import the folder '/path/to/import' into the workspace '/default-domain/workspaces/some-workspace' while monitoring the import logs from a REST client, use the following HTTP GET queries:

- `GET http://localhost:8080/nuxeo/site/fileImporter/logActivate`
- `GET http://localhost:8080/nuxeo/site/fileImporter/run?targetPath=/default-domain/workspaces/some-workspace&inputPath=/path/to/import&batchSize=10&interactive=false&nbThreads=`
- `GET http://localhost:8080/nuxeo/site/fileImporter/log`

In this section

To execute these HTTP queries you can either use a browser with an active Nuxeo session (JSESSIONID cookie) or use a third party stateless HTTP client with HTTP Basic Authentication, eg: with the curl command line client:

```
$ curl --basic -u 'Administrator:Administrator'
"http://localhost:8080/nuxeo/site/fileImporter/log"
```

Don't forget put the URL in quotes if it includes special shell characters such as '&'. You can also use the generic HTTP GUI client from the rest-client Java project: http://code.google.com/p/rest-client/
Be sure to fill in the 'Auth' tab with your user credentials.

Memory

The importer requires a lot of memory. Make sure your maximum heap size is set as high as possible for your environment. Maximum heap size can be set in nuxeo.conf in the JAVA_OPTS variable; for example, argument **-Xmx4g** will set maximum heap size to 4 gigabytes. See Configuration Parameters Index (nuxeo.conf) for more details.

REST API

Resource URL	Description	Output
`GET nuxeo/site/randomImporter/run`	Random text generator for load testing	text/plain; charset=UTF-8
`GET nuxeo/site/fileImporter/run`	Default file importer	text/plain; charset=UTF-8
`GET nuxeo/site/fileImporter/log`	Get current log buffer content	text/plain; charset=UTF-8
`GET nuxeo/site/fileImporter/logActivate`	Activate logging	text/plain; charset=UTF-8
`GET nuxeo/site/fileImporter/logDesactivate`	Deactivate logging	text/plain; charset=UTF-8
`GET nuxeo/site/fileImporter/status`	Get importer thread status	text/plain; charset=UTF-8 "Running" or "Not Running"

`GET nuxeo/site/fileImporter/kill`	Stop the importer thread if running	text/plain; charset=UTF-8

fileImporter/run

Parameter	Default value	Description
`leafType`	null	Leaf type used by the `documentModelFactory` for the import.
`folderishType`	null	Folderish type used by the `documentModelFactory` for the import.
`inputPath`	N/A	Root path to import (local to the server).
`targetPath`	N/A	Target path in Nuxeo
`skipRootContainerCreation`	false	If true the root container won't be created
`batchSize`	5	Number of documents that will be created before doing a commit
`nbThreads`	5	Maximum number of importer threads that can be allocated
`interactive`	false	

N/A: no default value, the parameter is required.

randomImporter/run

Parameter	Default value	Description
`targetPath`	N/A	Target path in Nuxeo
`skipRootContainerCreation`		
`batchSize`		Number of documents that will be created before doing a commit
`nbThreads`		Maximum number of importer threads that can be allocated
`interactive`		
`nbNodes`	N/A	Number of nodes to create
`fileSizeKB`		
`onlyText`	true	
`blockSyncPostCommitProcessing`		
`blockAsyncProcessing`		
`bulkMode`	true	

N/A: no default value, the parameter is required.

Extend

You can easily write your own importer, extending the `org.nuxeo.ecm.platform.importer.base.GenericMultiThreadedImporter` class.

Using XML extension points you can also define the different building blocks of the importer:

- class for reading source nodes,
- docType used for leaf Documents,
- docType used for folderish Documents,
- `documentModelFactoryClass`.

See the developer documentation of Nuxeo Bulk Document Importer for details.

See nuxeo-platform-importer Javadoc.

Directory Tree and Threading

The default importer is targeting a simple use case: import a complete filesystem tree inside a Nuxeo repository.

On most computers you have several CPUs and several cores: this means you can import more documents per second by using several threads. However, when importing a tree, threading must be considered carefully:

- Each thread will be associated with a Transaction (remember we import several documents before doing a commit),
- Each transaction is isolated from others (MVCC mode).

This means that a new thread must be created only when a new branch will be accessible inside the source filesystem. At least, the default `ImporterThreadingPolicy` (`DefaultMultiThreadingPolicy`) does that.

As a result, if you import a big folder with a flat structure, you will only have one importer thread, even if you configure to allow more.

To be sure to be able to leverage multi-threading, you can either:

- Ensure the source filesystem is a tree with at least two levels,
- Change the importer threading policy.

Importer and Metadata

The default importer provides two classes to read the source files as well as metadata:

`FileWithMetadataSourceNode`

This is the default implementation, that was mainly targeting at importing a filesystem where file are stored by folders.

The idea is to associate a set of metadata on a per folder basis: the `metadata.properties` will be used for defining the metadata for all files inside the same folder. By default, metadata will be inherited from parent folder, but may be completed or overridden by a local `metadata.properties`.

Here is a structure:

```
TopicA
   file1.pdf
   file2.pdf
   metadata.properties
   TopicA1
      file1.pdf
      file2.pdf
      metadata.properties
   TopicA2
      file1.pdf
      file2.pdf
      metadata.properties
   TopicA3
   file1.pdf
   file2.pdf
   metadata.properties
TopicB
file1.pdf
metadata.properties
TopicB1
   file1.pdf
   file2.pdf
   metadata.properties
TopicB12
file1.pdf
file2.pdf
metadata.properties
```

The `metadata.properties` file is a simple property file in the format `xpath = value`. Typically:

```
dc\:description=some desscription
dc\:source=some source
dc\:subjects=subject4|subject5
```

`FileWithIndividualMetadasSourceNode`

This second implementation will try to file a property file for each imported file. This allows to have a per file metadata set.

A sample structure would be:

```
branch1
   branch11
      hello11.pdf
      hello11.properties
   hello1.pdf
   hello1.properties
hello.pdf
hello.properties
```

To use this node type you need to redefine the importer. For that create a `importer-config.xml` in `nxserver/config`.

```
<?xml version="1.0"?>
<component name="customImporter">
<require>org.nuxeo.ecm.platform.importer.service.jaxrs.contrib</require>

<extension
target="org.nuxeo.ecm.platform.importer.service.DefaultImporterComponent"
point="importerConfiguration">
   <importerConfig sourceNodeClass
="org.nuxeo.ecm.platform.importer.source.FileWithIndividualMetadasSourceNode" >
      <documentModelFactory leafType="File" folderishType="Folder"
documentModelFactoryClass="org.nuxeo.ecm.platform.importer.factories.DefaultDocum
entModelFactory" />
    </importerConfig>
</extension>
</component>
```

Other Bulk Document Importer Documentation

- Bulk Document Importer (Nuxeo Installation and Administration - 5.8)
- Nuxeo Bulk Document Importer (Nuxeo Platform Developer Documentation - 5.8)

Digital Signature

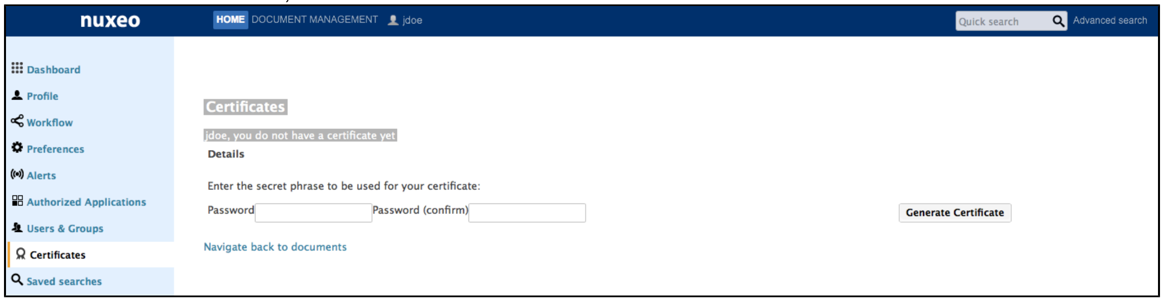
The Digital Signature addon introduces PDF signing capabilities to the Nuxeo Platform. This addon also provides generation of user certificates, which are required for document signing.

Installation

The Digital Signature package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, users get some new tabs:

- a Certificates tab in the Home,



Certificates

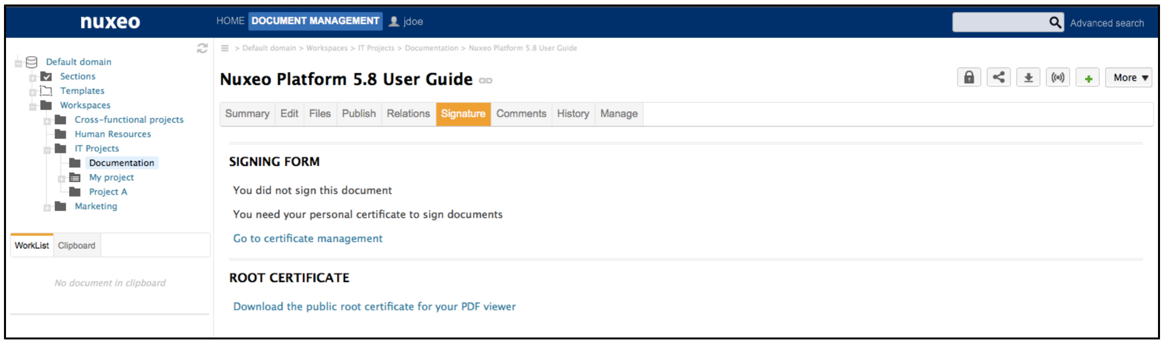
jdoe, you do not have a certificate yet

Details

Enter the secret phrase to be used for your certificate:

Password Password (confirm)

- a Signature tab on documents.



Nuxeo Platform 5.8 User Guide

SIGNING FORM

You did not sign this document

You need your personal certificate to sign documents

Go to certificate management

ROOT CERTIFICATE

Download the public root certificate for your PDF viewer

In this section

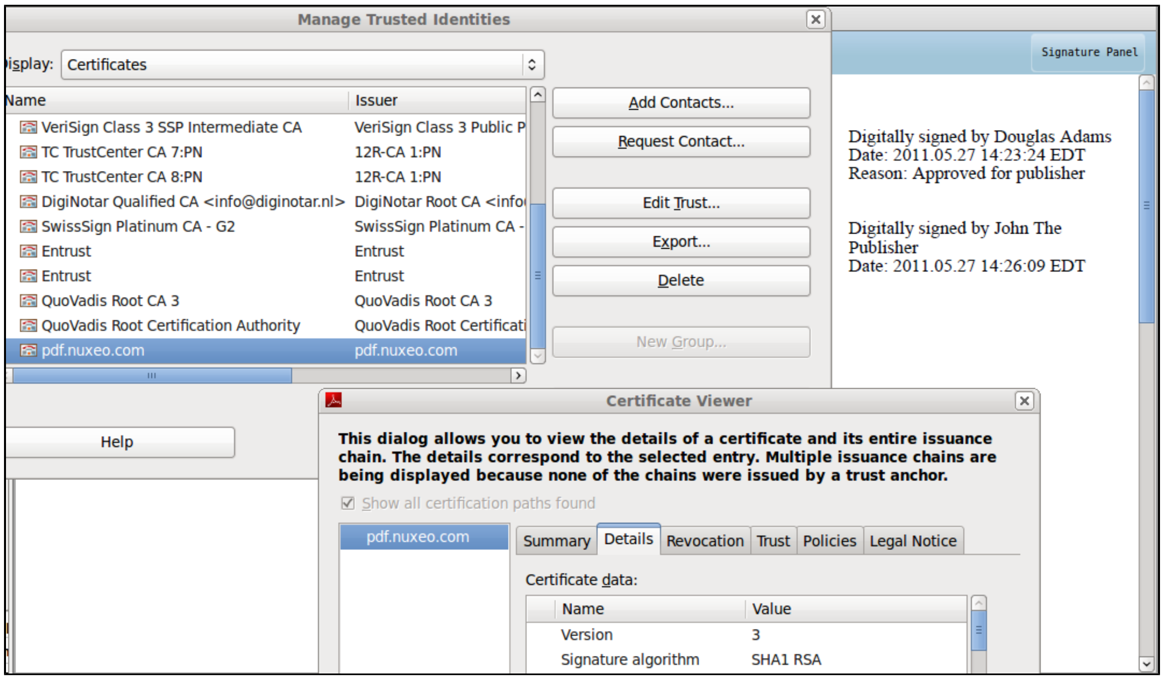
Configuration

The Digital Signature package provides a sample root certificate populated with sample company's values. However you will need to configure the package so that documents are signed with your company's certificate and information instead of the sample one.

Setting up the Local Root Certificate

To disambiguate, the term "root certificate" in this section - and in the configuration of this plugin - relates to the Local Certificate Authority (CA) of your company which is the root of all user certificates. This is not to be confused with the global root Certificate Authority, that is one of the top-most entities of the global "chain of trust".

This plugin's root certificate helps establishing a simple method of user certificate verification, as it can be installed in a PDF reader. The best approach, however, is to have your local Certificate Authority's certificate signed by a higher level CA whose ancestor has been signed by one of the actual root Certificate Authorities. This incurs some setup overhead in the initial stages of the project. This method guarantees, though, a more secure approach to document verification, and frees the end users from having to install certificates in their PDF readers. PDF readers capable of handling security are updated automatically with the global root Certificate Authority information.



Name	Issuer
VeriSign Class 3 SSP Intermediate CA	VeriSign Class 3 Public P
TC TrustCenter CA 7:PN	12R-CA 1:PN
TC TrustCenter CA 8:PN	12R-CA 1:PN
DigiNotar Qualified CA <info@diginotar.nl>	DigiNotar Root CA <info
SwissSign Platinum CA - G2	SwissSign Platinum CA -
Entrust	Entrust
Entrust	Entrust
QuoVadis Root CA 3	QuoVadis Root CA 3
QuoVadis Root Certification Authority	QuoVadis Root Certificati
pdf.nuxeo.com	pdf.nuxeo.com

Digitally signed by Douglas Adams
Date: 2011.05.27 14:23:24 EDT
Reason: Approved for publisher

Digitally signed by John The Publisher
Date: 2011.05.27 14:26:09 EDT

This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry. Multiple issuance chains are being displayed because none of the chains were issued by a trust anchor.

☑ Show all certification paths found

pdf.nuxeo.com

Certificate data:

Name	Value
Version	3
Signature algorithm	SHA1 RSA

As the keystore configured in the installable package is a sample keystore containing a test configuration, it is required that it be replaced with the client keystore containing the keypair and the certificate to be used for signing user certificates. As of now the certificate+keypair need to be stored in a .jks formatted keystore and configured via the extension mechanism.

The user certificate generation step requires a Certificate Authority certificate (CA) to be set up inside the Nuxeo Platform system as all user certificates have to be signed by a CA with a recognizable identity — a company rather than a single user. The term local CA can be understood here as "company Certificate Authority" or "system-wide Certificate Authority". Note that there is only one CA certificate per system but each user can have his own certificate.

Setting up a CA certificate from a 3rd party authority

For this exercise you will need the following software:

keytool: the keytool comes with your JDK (Java Development Kit) installation.

openssl: Open SSL

1. Create a keypair (with alias `pdfcakey` in this example).

   ```
   keytool -genkey -keyalg RSA -alias pdfcakey -keypass password -validity 365
   -keysize 1024 -dname "cn=PDF-CA, ou=Headquarters, o=Example Organization,
   c=US" -keystore pdfca-keystore.jks
   ```

   This creates a keypair (private and public key), and self-signs it automatically.

   If you don't wish to use a 3rd party Certificate Authority to sign your key, you can stop here.
2. Create a certificate signing request (CSR).

   ```
   keytool -keystore pdfca-keystore.jks -storepass aaaaaa -alias alternatekey
   -keypass password -certreq -file pdfca.csr
   ```

3. Submit the CSR to a well-known 3rd party Certificate Authority of your choice to sign it.

   You can find examples of 3rd party CAs here and here.
4. When you receive the signed certificate pdfca.crt, import it into your keystore using a new new alias (`pdfcacert` in this example).

   ```
   keytool -import -trustcacerts -alias pdfcacert -file pdfca.crt -keystore
   pdfca-keystore.jks
   ```

Setting up a local CA certificate

An alternative method would be to set up a local signing CA and use it for signing certificates.

⚠ Though it could work for small-scale deployments, this approach is not recommended for production purposes.

Step 1: Create a Certificate Authority (CA)

1. Create a CA key.

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Create a self signed CA certificate.

   ```
   openssl req -new -x509 -days 356 -key ca.key -out ca-self-signed.crt
   ```

Step 2: Create Subordinate Certificate Authority (SUBCA)

1. Create the key for the subordinate CA.

   ```
   openssl genrsa -out subca.key 2048
   ```

2. Create a certificate signing request (CSR) for the subordinate CA.

   ```
   openssl req -new -key subca.key -out subca.csr
   ```

3. Sign the CSR of the subordinate CA.

   ```
   openssl x509 -req -days 730 -in subca.csr -CA ca-self-signed.crt -CAkey
   ca.key -set_serial 01 -out subca.crt
   ```

4. Import a certificate created from your CSR into a JKS keystore.

   ```
   keytool -import -alias certalias -file subca.crt -keystore keystore.jks
   ```

5. Convert the x509-certificate and the key to pkcs12 format to make it importable into the java keystore.

   ```
   openssl pkcs12 -export -in subca.crt -inkey subca.key -name keyalias
   -CAfile ca.crt -caname root -out subca.p12
   ```

   (use "export" as password when prompted)

6. Convert the pkcs12 file to JKS format.

```
keytool -importkeystore -deststorepass storepass -destkeypass keypass
-destkeystore keystore.jks -srckeystore subca.p12 -srcstoretype PKCS12
-srcstorepass export -alias keyalias
```

   Now you will need to replace the sample certificate with your own that you just created. You can use the configuration information below which explains how to override the sample certificate with your company certificate.

Step 3: Replace the sample root certificate

1. Create a `***-config.xml` (e.g.`rootcert-digitalsignature-config.xml`) file with the content below:

```
<component name="my.signature.rootservice.config">
  <require>org.nuxeo.signature.config.default</require>
  <extension target="org.nuxeo.ecm.platform.signature.api.pki.RootService"
point="rootconfig">
    <configuration>
      <rootKeystoreFilePath>test-config/keystore.jks</rootKeystoreFilePath>
      <rootKeystorePassword>abc</rootKeystorePassword>
      <rootCertificateAlias>pdfcacert</rootCertificateAlias>
      <rootKeyAlias>pdfcakey</rootKeyAlias>
      <rootKeyPassword>abc</rootKeyPassword>
    </configuration>
  </extension>
</component>
```

2. Put the extension in the `config` directory of your server:
   - `$NUXEO/nxserver/config` for a Tomcat distribution,
   - `$NUXEO/server/default/deploy/nuxeo.ear/config` for a JBoss distribution.

Setting up the Company Information for New Certificates

Another extension provides general company information used in all certificates, like Country, Locale, Organization Name and Organizational Unit.

To add your company's information for users certificates:

1. Create another XML file called `***-config.xml` (e.g.`companyinfo-digitalsignature-config.xml`) with the content below:

```
<?xml version="1.0"?>
<component name="my.signature.userservice.config">
  <require>org.nuxeo.signature.config.default</require>
  <extension
target="org.nuxeo.ecm.platform.signature.api.user.CUserService"
point="cuserdescriptor">
    <userDescriptor>
      <countryCode>MX</countryCode>
      <organization>Sigma Alimentos</organization>
      <organizationalUnit>Marketing</organizationalUnit>
    </userDescriptor>
  </extension>
</component>
```

2. Put the extension in the `config` directory of your server:
   - `$NUXEO/nxserver/config` for a Tomcat distribution,
   - `$NUXEO/server/default/deploy/nuxeo.ear/config` for a JBoss distribution.

Other documentation about this package

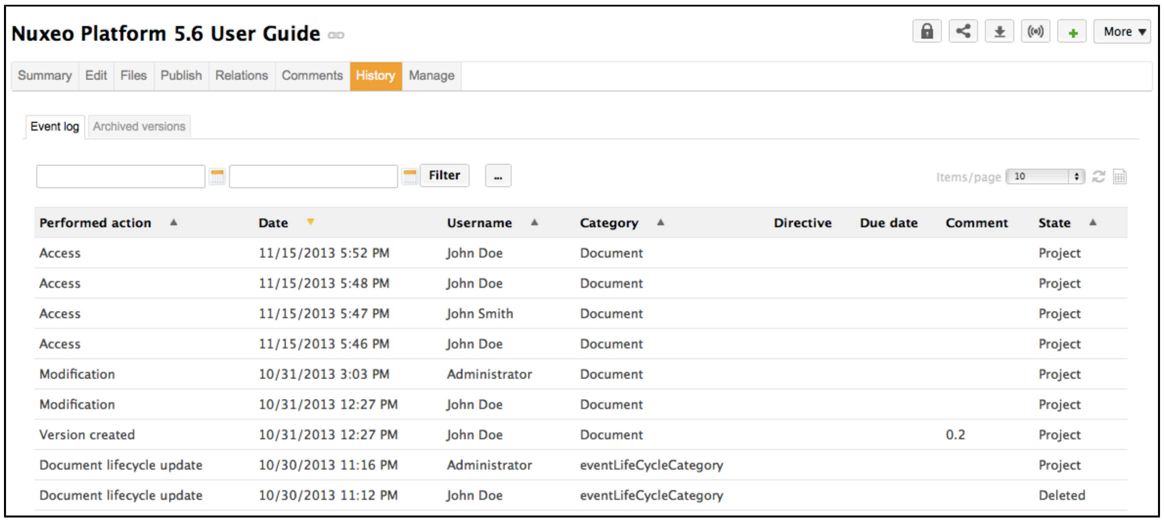
- Digital Signature (Nuxeo Platform Developer Documentation - 5.8)
- Digital Signature (Nuxeo Platform User Documentation - 5.8)

Document Access Tracking

The Document access tracking package is used to register in the document's history the fact that users have accessed the document, and so have probably read it.

The Document access tracking requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After you installed the package, an "Access" action is logged in the History tab of documents every time a user click on it to consult it.



Performed action	Date	Username	Category	Directive	Due date	Comment	State
Access	11/15/2013 5:52 PM	John Doe	Document				Project
Access	11/15/2013 5:48 PM	John Doe	Document				Project
Access	11/15/2013 5:47 PM	John Smith	Document				Project
Access	11/15/2013 5:46 PM	John Doe	Document				Project
Modification	10/31/2013 3:03 PM	Administrator	Document				Project
Modification	10/31/2013 12:27 PM	John Doe	Document				Project
Version created	10/31/2013 12:27 PM	John Doe	Document			0.2	Project
Document lifecycle update	10/30/2013 11:16 PM	Administrator	eventLifeCycleCategory				Project
Document lifecycle update	10/30/2013 11:12 PM	John Doe	eventLifeCycleCategory				Deleted

Other documentation about this package

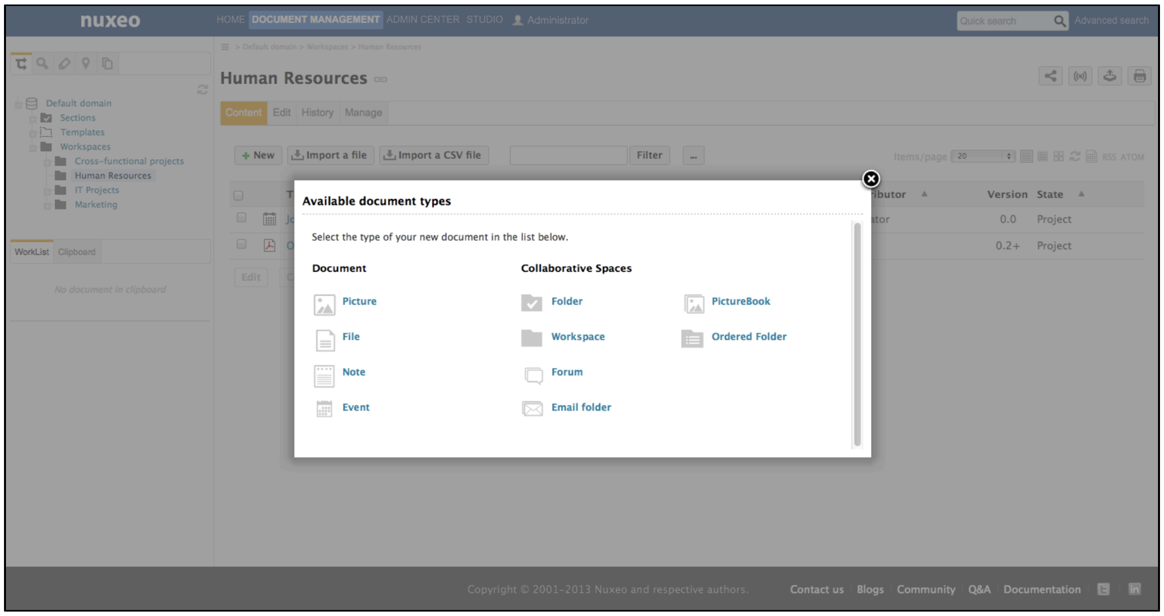
- Document access tracking (Nuxeo Platform User Documentation - 5.8)

Nuxeo Agenda

The Nuxeo Agenda package provides users with a new documents type "Event" that enables them to manage their list of meetings and other events with a calendar view.

The Nuxeo Agenda package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, a new document type is available from workspaces, called "Event" and users can add an Agenda gadget on their dashboard.



Other documentation about this addon

Nuxeo Agenda (Nuxeo Platform User Documentation - 5.8)

Nuxeo - BIRT Integration

The Nuxeo - BIRT Integration package leverages the reporting features of Eclipse BIRT, enabling users to create reports on the application's activity, directly from the Nuxeo Platform.

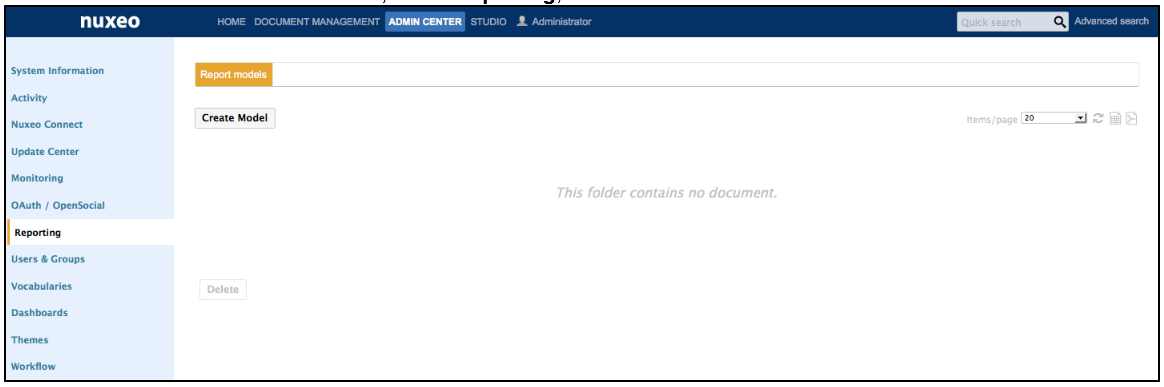
Installation

The Nuxeo - BIRT Integration addon requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center. Note however that it requires the Document Management module, and that your Nuxeo application must run with PostgreSQL.

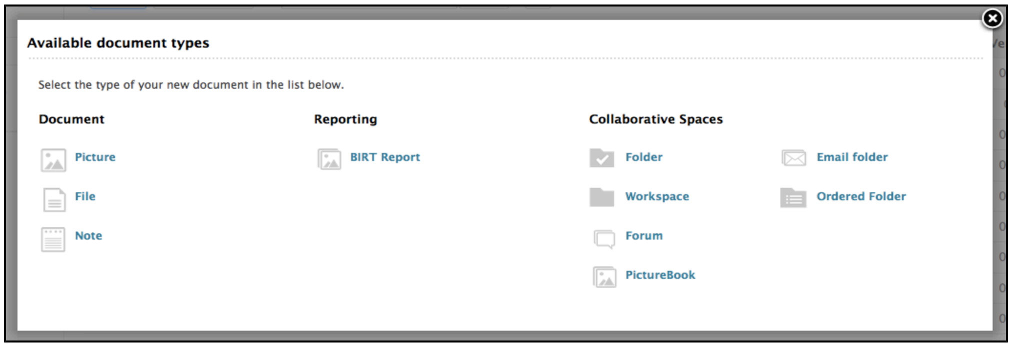
After you installed the Nuxeo - BIRT Integration package, here are the changes you get in the Nuxeo Platform:

In this section

- the Admin Center has a new vertical tab, called **Reporting**;



- A new document type is available in workspaces and sections, called **BIRT Report**.



Configuration

The Nuxeo - BIRT Integration addon requires that your Nuxeo application runs with PostgreSQL. See the Connecting Nuxeo to the Database page.

Other documentation about Nuxeo - BIRT Integration

Nuxeo - BIRT Integration (Nuxeo Platform User Documentation - 5.8)

Nuxeo CSV

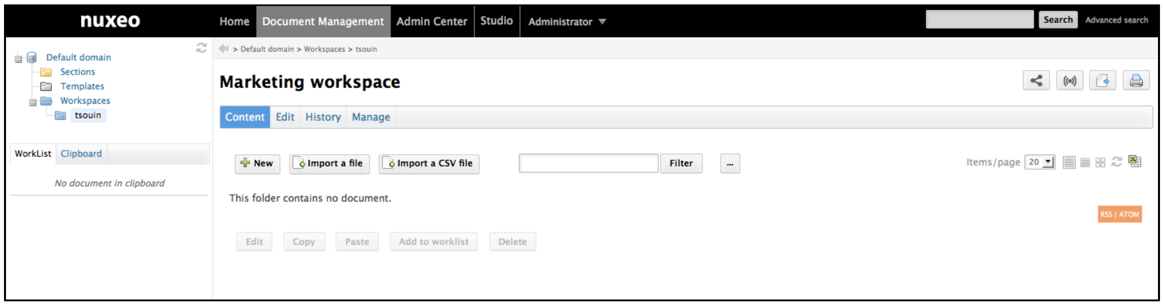
The Nuxeo CSV addon enables users to proceed to a bulk import of documents in the Nuxeo Platform using a CSV file. This addon enables to create documents with their metadata filled in, to import files with their main attachment, to create a tree structure.

In this section

Installation

The Nuxeo CSV package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, users have a **Import a CSV file** button available in workspaces, folders and in any document where they can import files.



Configuration: Enabling File Upload

The Nuxeo CSV addon enables users to create file documents and upload their main attachment at the same time. This requires to configure where the server will take the attachments. This is done adding the parameter `nuxeo.csv.blobs.folder` in the server nuxeo.conf and giving him a value that is the path to a folder that can be accessed by the server.

Other pages about Nuxeo CSV

Nuxeo CSV: Enabling CSV import on a Custom Document Type (Nuxeo Platform Developer Documentation - 5.8)

Nuxeo CSV (Nuxeo Platform User Documentation - 5.8)

Error rendering macro 'contentbylabel' : com.atlassian.confluence.macro.params.ParameterException: 'NXDOC5858' is not an existing space's key

Nuxeo DAM

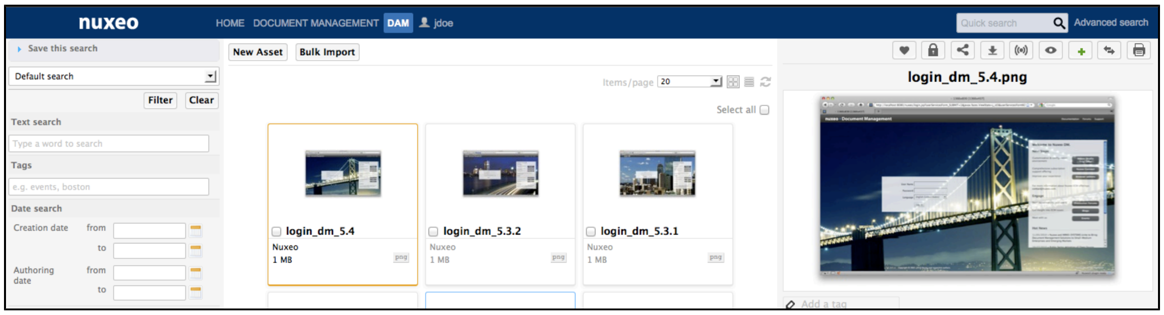
The Digital Asset Management module of the Nuxeo Platform provides organizations with an application to manage their digital assets which can be office (MS, OOo, PDF, ...) or multimedia (pictures, audio and video) files.

The main difference with Nuxeo DM (Document Management) is that DAM is not aimed at "producing" new documents with collaborative editing / review workflows but merely at browsing an existing collection that have been authored externally and imported in batch in the DAM application.

User will merely edit the properties / categories or add annotations to enrich the asset without editing the main attached file. The main edit actions would preferably take place in the Document Management UI.

The Digital Asset Management module requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center. It is also possible to install it from the Startup wizard.

After the package is installed, users have a DAM tab on top of the application, next to the Document Management tab.



Other documentation about DAM

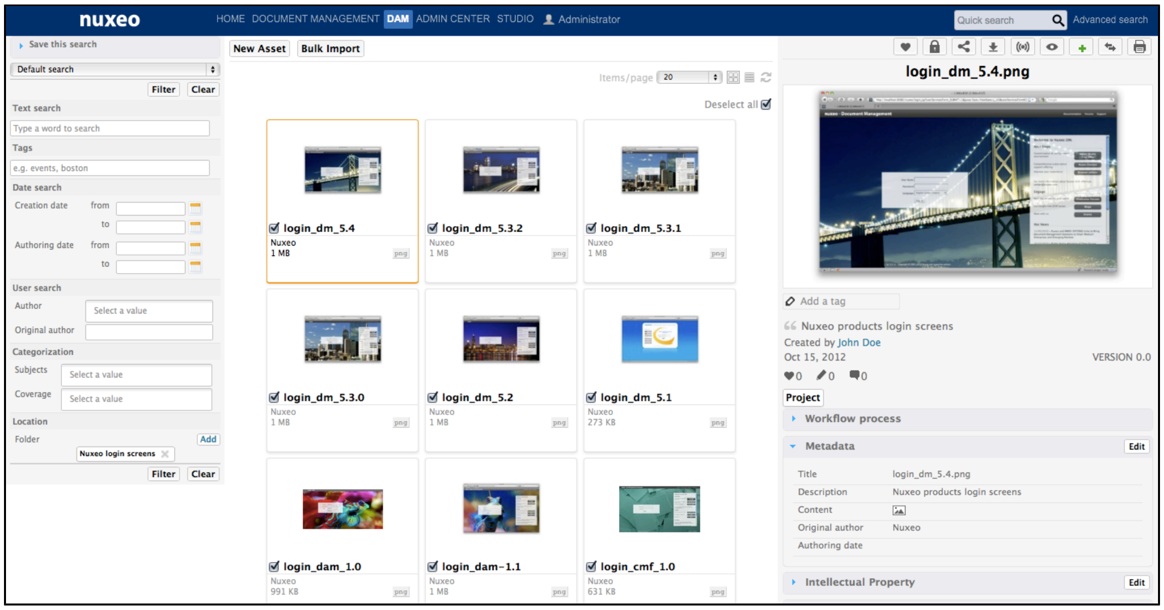
- Nuxeo DAM (Nuxeo Installation and Administration - 5.8)
- Digital Asset Management (DAM) (Nuxeo Platform Developer Documentation - 5.8)
- Digital Asset Management (Nuxeo Platform User Documentation)
- Customizing the new Bulk Import UI (Nuxeo Platform Developer Documentation - 5.8)
- User actions categories (Nuxeo Studio)

Nuxeo DAM PDF Export

The Nuxeo DAM PDF Export package enables users to export a selection of pictures in a PDF document.

The Nuxeo DAM PDF Export package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, a new **Export as PDF** button is displayed at the bottom of the documents lists in the DAM main tab.



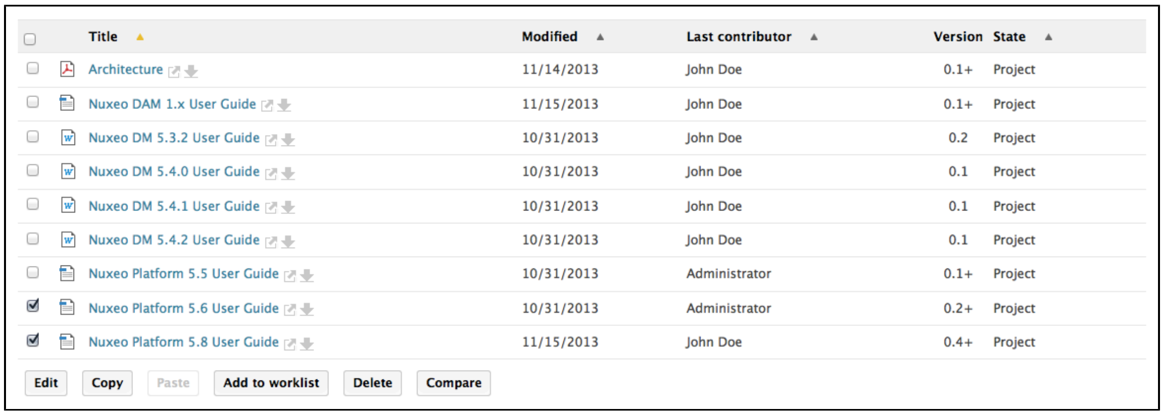
Error rendering macro 'contentbylabel' : com.atlassian.confluence.macro.params.ParameterException: 'NXDOC5858' is not an existing space's key

Nuxeo Diff

Nuxeo Diff enables to compare two documents or two versions of a document to see the differences between documents or versions.

The Nuxeo Diff package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After Nuxeo Diff is installed, a new **Compare** button is available in workspaces and in the document's **History** > **Archived versions** tab.



	Title	Modified	Last contributor	Version	State
☐	Architecture	11/14/2013	John Doe	0.1+	Project
☐	Nuxeo DAM 1.x User Guide	11/15/2013	John Doe	0.1+	Project
☐	Nuxeo DM 5.3.2 User Guide	10/31/2013	John Doe	0.2	Project
☐	Nuxeo DM 5.4.0 User Guide	10/31/2013	John Doe	0.1	Project
☐	Nuxeo DM 5.4.1 User Guide	10/31/2013	John Doe	0.1	Project
☐	Nuxeo DM 5.4.2 User Guide	10/31/2013	John Doe	0.1	Project
☐	Nuxeo Platform 5.5 User Guide	10/31/2013	Administrator	0.1+	Project
☑	Nuxeo Platform 5.6 User Guide	10/31/2013	Administrator	0.2+	Project
☑	Nuxeo Platform 5.8 User Guide	11/15/2013	John Doe	0.4+	Project

Other Nuxeo Diff documentation

Nuxeo Diff (Nuxeo Platform Developer Documentation - 5.8)

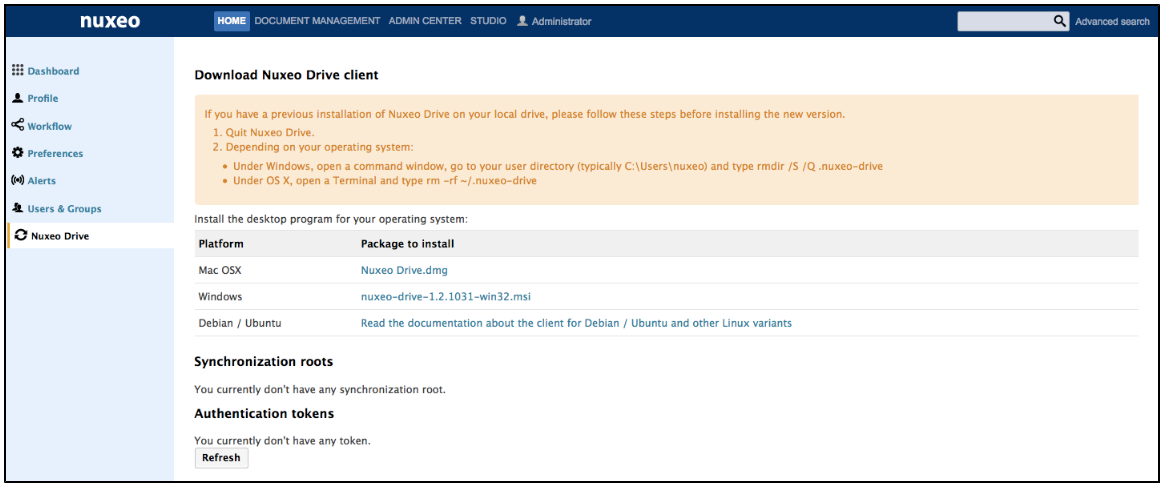
Nuxeo Diff (Nuxeo Platform User Documentation - 5.8)

Nuxeo Drive

Nuxeo Drive is a Nuxeo addon that enables the synchronization of folders or workspaces from the Nuxeo Platform with local folder on your computer.

The Nuxeo Drive package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After Nuxeo Drive has been installed on the server, users have a Nuxeo Drive tab on their Home.



Download Nuxeo Drive client

If you have a previous installation of Nuxeo Drive on your local drive, please follow these steps before installing the new version.

1. Quit Nuxeo Drive.
2. Depending on your operating system:
   - Under Windows, open a command window, go to your user directory (typically C:\Users\nuxeo) and type rmdir /S /Q .nuxeo-drive
   - Under OS X, open a Terminal and type rm -rf ~/.nuxeo-drive

Install the desktop program for your operating system:

Platform	Package to install
Mac OSX	Nuxeo Drive.dmg
Windows	nuxeo-drive-1.2.1031-win32.msi
Debian / Ubuntu	Read the documentation about the client for Debian / Ubuntu and other Linux variants

Synchronization roots

You currently don't have any synchronization root.

Authentication tokens

You currently don't have any token.

Refresh

Other documentations about Nuxeo Drive

Nuxeo Drive (Nuxeo Platform Developer Documentation - 5.8)

Nuxeo Drive (Nuxeo Platform User Documentation - 5.8)

How to Manually Initialize a Nuxeo Drive Instance

Usually Nuxeo Drive is automatically initialized at first startup. This includes:

- The creation of the configuration folder: `~/.nuxeo-drive`
- The creation of the local folder: `~/.Nuxeo Drive`
- The initialization of the SQLite database: `~/.nuxeo-drive/nxdrive.db`

However, you might want to do this initialization manually, for example to preset the Nuxeo server URL and proxy configuration before launching Nuxeo Drive for the first time. This can be useful for the deployment of Nuxeo Drive on a large set of desktops, allowing end users to work on a preconfigured instance, only needing to provide their credentials at first startup.

On this page

Please note that we only provide UNIX command lines in the following process, but they can easily be adapted for Windows. Of course all this can be scripted.

Configuration Folder and SQLite Database File Creation

```
mkdir ~/.nuxeo-drive
touch ~/.nuxeo-drive/nxdrive.db
```

Device Id Generation

The `device_config` table of the SQLite database needs a unique id as a primary key of its single row (`device_id` column). You first need to generate this id, for example with Python:

```
ataillefer@taillefer-xps:~$ python
    Python 2.7.3 (default, Sep 26 2013, 20:03:06)
    [GCC 4.6.3] on linux2
    Type "help", "copyright", "credits" or "license" for more information.
    >>> import uuid
    >>> uuid.uuid1().hex
    '1bd6686882c111e391a6c8f733c9742b'
    >>> exit()
```

SQLite Database Initialization

1. Connect to the empty SQLite database.

```
sqlite3 ~/.nuxeo-drive/nxdrive.db
```

2. Create the `device_config` and `server_bindings` tables.

```
CREATE TABLE device_config (
            device_id VARCHAR NOT NULL,
            client_version VARCHAR,
            proxy_config VARCHAR,
            proxy_type VARCHAR,
            proxy_server VARCHAR,
            proxy_port VARCHAR,
            proxy_authenticated BOOLEAN,
            proxy_username VARCHAR,
            proxy_password BLOB,
            proxy_exceptions VARCHAR,
            PRIMARY KEY (device_id),
            CHECK (proxy_authenticated IN (0, 1))
        );
        CREATE TABLE server_bindings (
            local_folder VARCHAR NOT NULL,
            server_url VARCHAR,
            remote_user VARCHAR,
            remote_password VARCHAR,
            remote_token VARCHAR,
            last_sync_date INTEGER,
            last_ended_sync_date INTEGER,
            last_root_definitions VARCHAR,
            PRIMARY KEY (local_folder)
        );
```

3. Insert the single row in `device_config`.

   Use the previously generated id for the `device_id` column, and set your proxy settings as in the example below.

```
INSERT INTO device_config (device_id, proxy_config, proxy_type,
proxy_server, proxy_port, proxy_authenticated) VALUES
('1bd6686882c111e391a6c8f733c9742b', 'Manual', 'http', '10.218.9.82',
'80', 0);
```

4. Insert a row in `server_bindings` .

Use your local folder path and the Nuxeo server URL.

```
INSERT INTO server_bindings (local_folder, server_url) VALUES
('/home/ataillefer/Nuxeo Drive', 'http://10.214.4.90:8080/nuxeo/');
```

5. Quit SQLite.

```
.exit
```

Start Nuxeo Drive

The Settings popup should appear waiting for the user's credentials only.

Other documentation about Nuxeo Drive

- Nuxeo Drive (Nuxeo Installation and Administration - 5.8)
- How to Manually Initialize a Nuxeo Drive Instance (Nuxeo Installation and Administration - 5.8)
- Nuxeo Drive (Nuxeo Platform Developer Documentation - 5.8)
- Nuxeo Drive (Nuxeo Platform User Documentation - 5.8)

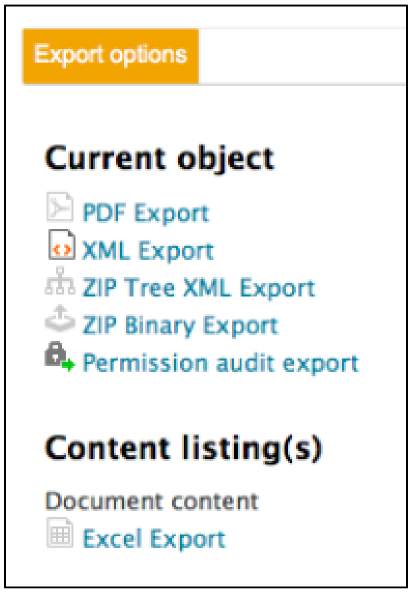
Nuxeo Groups and Rights Audit

The Nuxeo Groups and Rights Audit addon generates an Excel matrix listing every exported documents with permissions for each user.

Installation

The Nuxeo Groups and Rights Audit package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After it has been installed, administrators have a new export option available, called "Permission audit export".



In this section

Configuration

Setting Up e-Mail Sending

The Nuxeo Groups and Rights Audit addon sends email to the administrator who requested the audit. So your Nuxeo server must be able to reach an e-mail server. This is the same configuration that the one required for the email alerts to work. See how to enable e-mail alerts.

Setting Up a Higher Timeout

The default timeout to process the export of rights is 1200 seconds (20 minutes). You can change this default timeout by adding the parameter `nuxeo.audit.acl.timeout` to the nuxeo.conf file and defining another value than 1200, like 3600 (1 hour) for instance.

Other documentation about Nuxeo Groups and Rights Audit

Nuxeo Groups and Rights Audit (Nuxeo Platform User Documentation - 5.8)

Nuxeo jBPM

The Nuxeo jBPM package was last released in Fast Track version 5.7.2. Please check its latest LTS (5.6) documentation.

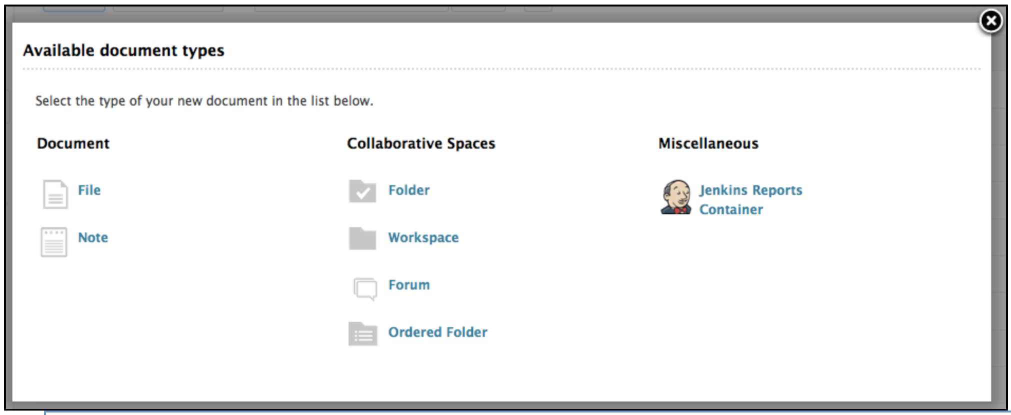
Nuxeo Jenkins Report

The Nuxeo Jenkins Report addon enables users of the Nuxeo Platform to generate and send reports on the status of the Continuous Integration on Jenkins, directly from the Nuxeo Platform. This addon is for development teams, to help them follow and share the status of their continuous integration, while leveraging the content management features of the Nuxeo Platform.

Installation

The Nuxeo Jenkins Report addons package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After you installed the package, a new document type is available for creation in workspaces and reports: the Jenkins Reports Container.



In this section

Configuration

Since this addon enables users to send the report from the Nuxeo Platform, your Nuxeo server must be able to reach an e-mail server. This is the same configuration that the one required for the email alerts to work. See how to enable e-mail alerts.

Error rendering macro 'contentbylabel' : null

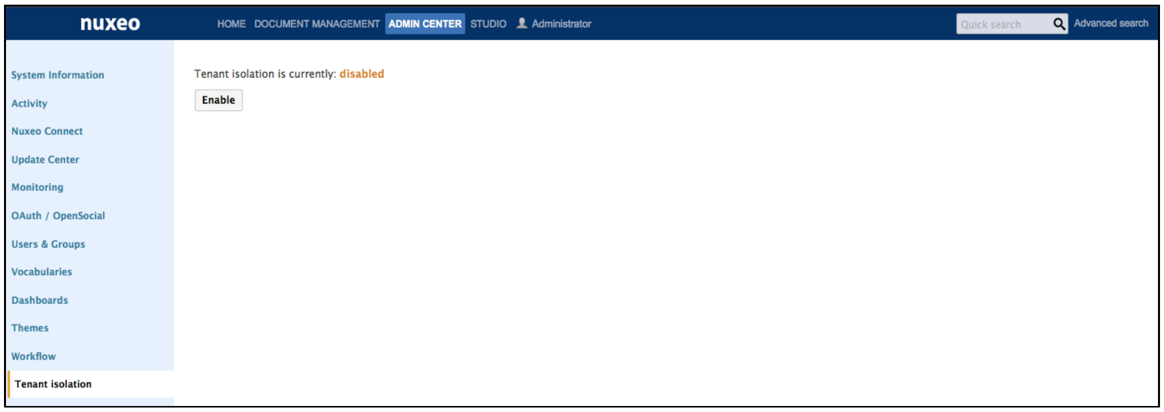
Nuxeo Multi-Tenant

The Multi-tenant addon enables to have domains, or tenants, that are independent from each other, with their own users, vocabulary values etc.

Installation

The Nuxeo Multi-tenant package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After you installed it, a tab **Tenant isolation** is available in the Admin Center.



Tenant isolation is currently: disabled

Other documentation about Nuxeo Multi-tenant

Nuxeo Multi-Tenant (Nuxeo Platform User Documentation - 5.8)

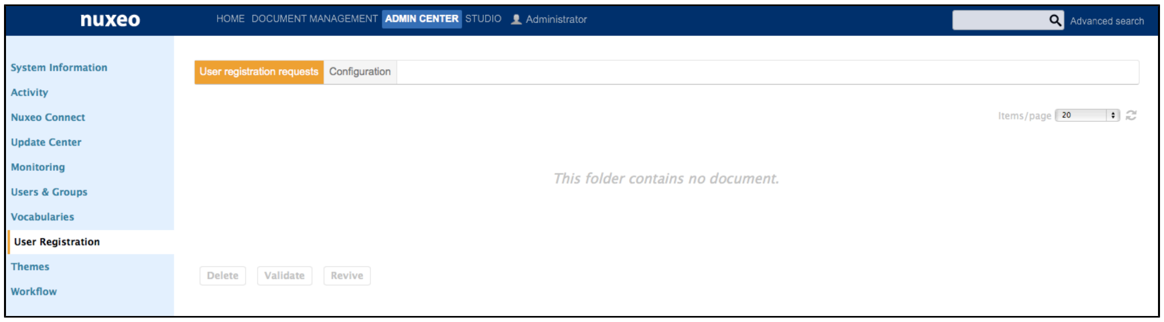
Nuxeo Platform User Registration

The Nuxeo Platform User Registration addon enables users to invite external users to access a specific space of the Platform or a limited set of spaces. The invitations must be approved by an administrator of the Platform.

Installation

The Nuxeo Platform User Registration package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package has been installed, a new User Registration tab is available in the Admin Center.

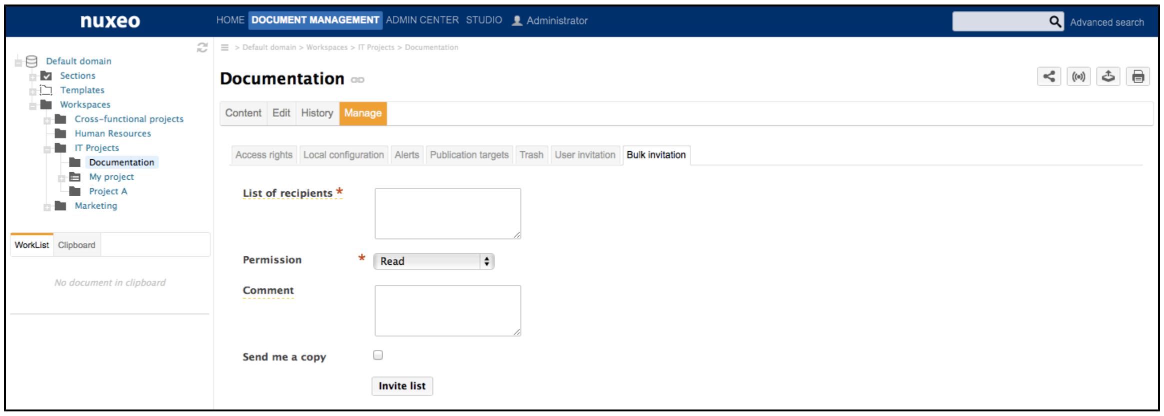


This folder contains no document.

In this section

Users with Manage rights on workspaces and section have two new subtabs in the Manage tab:

- User invitation
- Bulk invitation



List of recipients *	
Permission *	Read
Comment	
Send me a copy	☐

Configuration

Setting Up Email Sending

The Nuxeo Platform User Registration addon sends email to the invited user with his credentials. So your Nuxeo server must be able to reach an e-mail server. This is the same configuration that the one required for the email alerts to work. See how to enable e-mail alerts.

Global Configuration

The administrators can set up some configuration directly from the **Admin Center** > **User registration** > **Configuration** tab.

Possible configuration options are:

Field	Description
Allow new user creation	Enables users to invite user that don't have an account on the Platform. A new user account is then created. The new user is not included in any group by default.
Force rights assignment	This option is useful when user is manually created or comes from another system.
Direct validation if user exists	If a user invites a user that has already been invited to another space and so who already has a account on the Platform, then the administrators don't have to approve the invitation again. It is directly approved by the system.
Local registration tab	Displays a User registration requests subtab in the local Manage tab of a space, that displays the invitations that were done from the current space and their status.

Other documentation about this addon

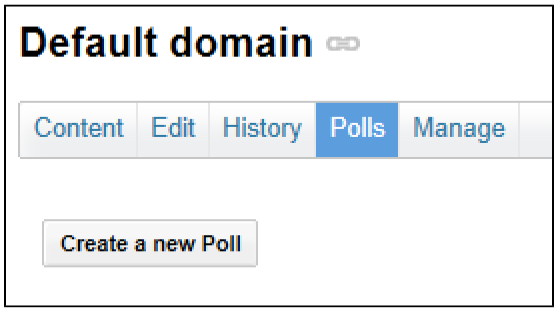
Nuxeo Platform User Registration (Nuxeo Platform User Documentation - 5.8)

Nuxeo Poll

The Nuxeo Poll package enables Nuxeo Platform users to create surveys and have a visual overview of the results.

There are no specific requirements to install the Nuxeo Poll package. You can install it from the Marketplace or from the Admin Center.

To check that the Nuxeo Poll package was correctly installed, browse the Nuxeo Platform with the Administrator user: you can see that workspaces, sections ans templates now have an additional tab, called **Polls**.



Other Documentation about This Package

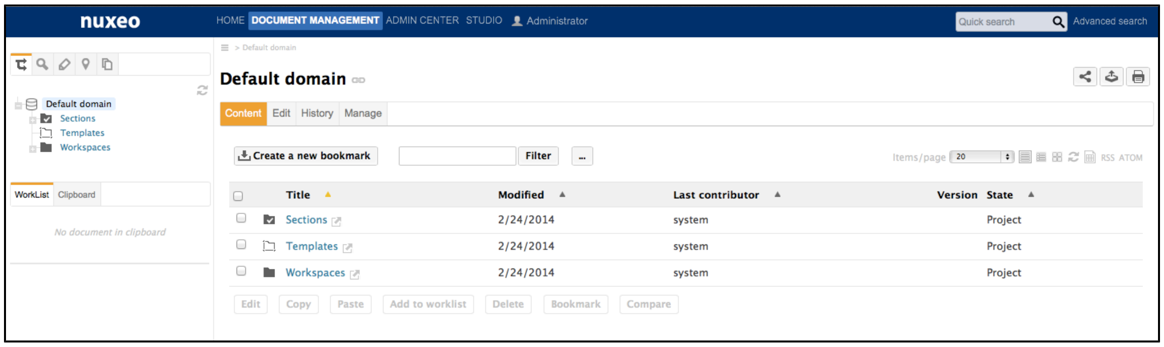
Nuxeo Poll (Nuxeo Platform User Documentation - 5.8)

Nuxeo Shared Bookmarks

The Nuxeo Shared Bookmarks addon enables users to bookmark documents and organize their bookmarks in folders. They can thus organize existing documents in a new tree structure without duplicating content.

The Nuxeo Shared Bookmarks addon requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, a new button **Bookmark** button is available next to the Add to worklist, Copy, Delete buttons, and a **Create a new bookmark** button.



Title	Modified	Last contributor	Version	State
Sections	2/24/2014	system		Project
Templates	2/24/2014	system		Project
Workspaces	2/24/2014	system		Project

Related pages in current documentation

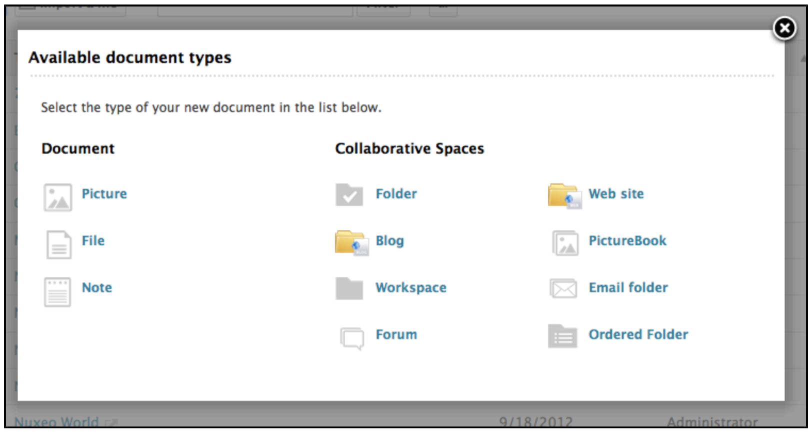
Nuxeo Shared Bookmarks (Nuxeo Platform User Documentation - 5.8)

Nuxeo Sites and Blogs

The Nuxeo Sites and Blogs package provides two new documents types to the Platform: websites and blogs. Websites and Blogs are collaborative documents that are web publishing oriented. As so, they have a second interface that makes it easy to display the documents of a workspace to the public. These specific presentations are built using Nuxeo WebEngine.

The Nuxeo Sites and Blogs package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, two new document types are available when you click on the **New** button: Websites and Blogs.



Other documentation about the Nuxeo Sites and Blogs package

- Nuxeo Sites and Blogs (Nuxeo Platform User Documentation - 5.8)

Resources Compatibility

The Resources Compatibility add-on provides backward compatibility with web resources (icons, JavaScript, ...) that have been removed from the previous LTS release of Nuxeo Platform.

The Resources Compatibility add-on requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

Other documentation about this addon

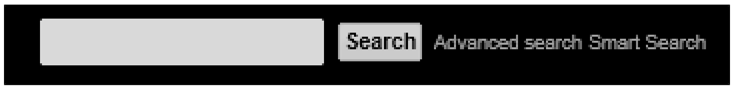
- Resources Compatibility (Nuxeo Platform Developer Documentation - 5.8)

Smart Search

The Smart Search package is a query engine that adds a new search form in the application from which you can build your queries and save them in smart folders. It offers search criteria on content, dates, and metadata.

The Smart search package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, a new Smart search link is available in the top right corner of the application.



Other documentation about this package

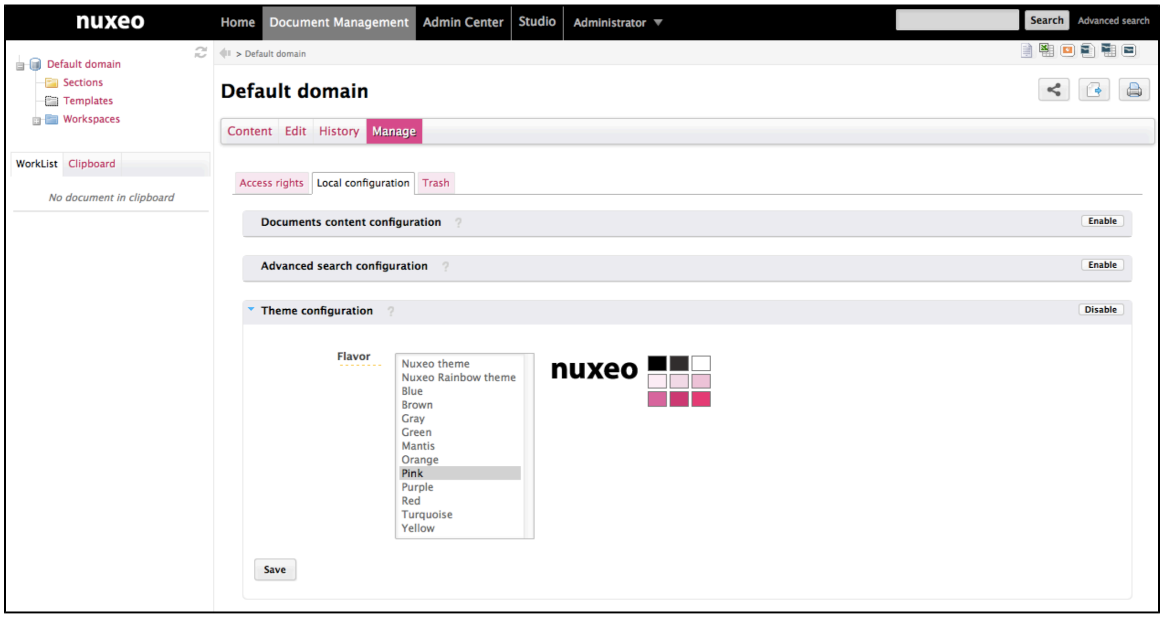
- Smart Search (Nuxeo Platform Developer Documentation - 5.8)
- Smart search operators (Nuxeo Platform User Documentation - 5.8)
- Smart Search (Nuxeo Platform User Documentation - 5.8)

Unicolor Flavors Set

This Unicolor Flavors Set package lists a set of flavors that customizes the colors of your workspaces, sections or any space on your Nuxeo application.

The Unicolor Flacors Set package requires no specific installation steps. It can be installed like any other package from the Marketplace or from the Admin Center.

After the package is installed, new flavors are available in workspaces Theme configuration.



Other documentation about Unicolor Flavors Set

Unicolor Flavors Set (Nuxeo Platform User Documentation - 5.8)