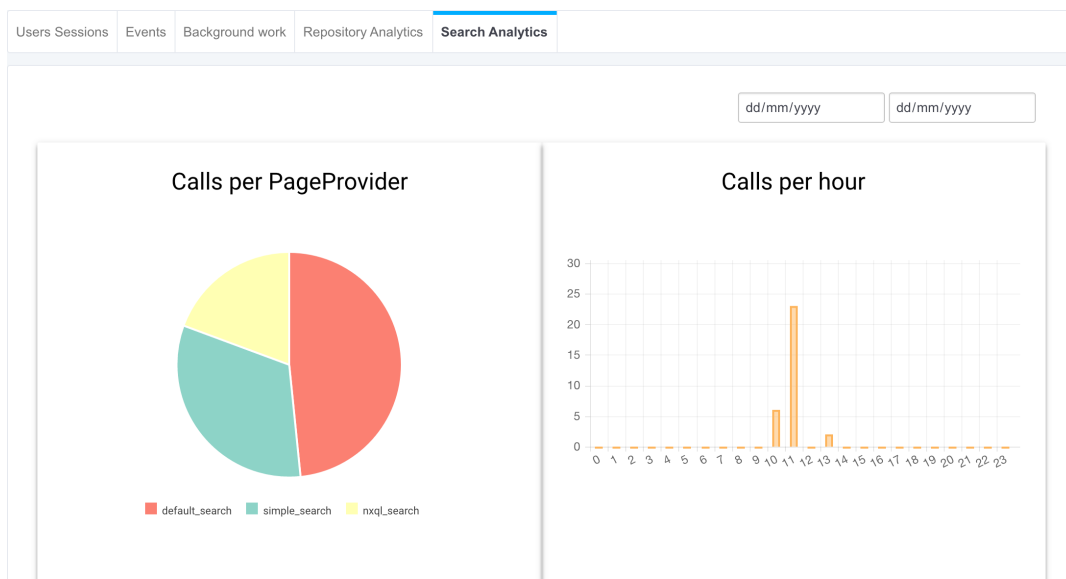


Nuxeo Platform 7.10 Release Notes

For Users

Search Analytics

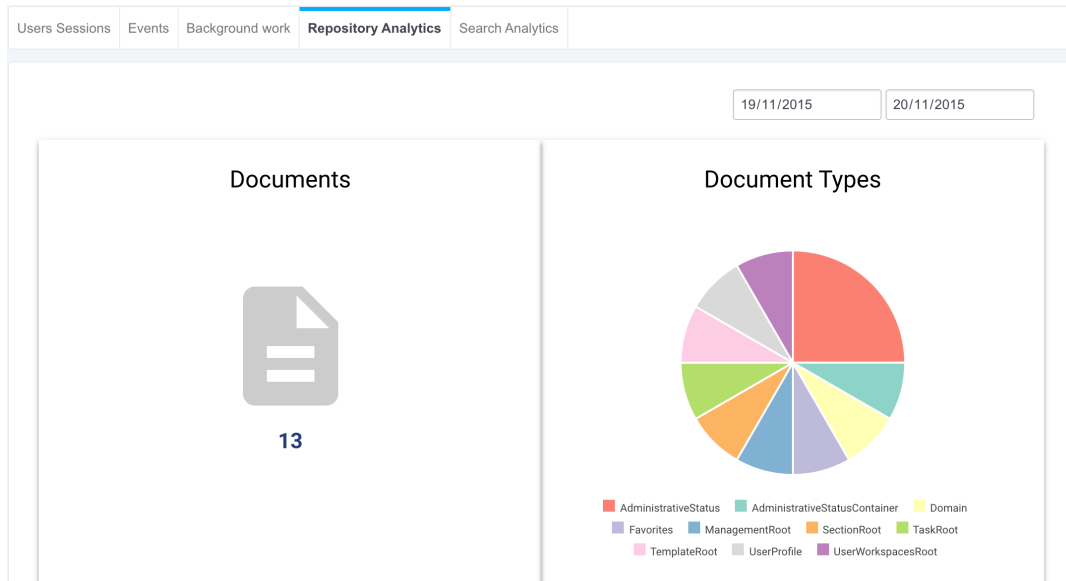
A new Dataviz dashboard has been added in the Admin Center for understanding search usages of your application. Search Screens usage, time of the day where it is the most used, average number of documents per result, most frequent terms used in full text search, which field is the most used. This dashboard is compatible with the customisation you would do using Nuxeo Studio.



Repository Analytics

In the Admin Center/Activity tab, a new tab "Repository Analytics" makes use of the nuxeo-repository-data element to display information about the content. With a between dates selector: (no selection means of all times)

- Number of documents (gauge)
- Pie Chart per document type
- Top 10 creators Pie Chart
- Documents creation per week
- Documents modification per week



More Robust Picture Diff

[Nuxeo Diff Picture](#) is now able to compare pictures that represent the same object but with different dimensions and/or picture format, leveraging Image Magick capabilities in this area.

Tabs Are Ajaxified by Default

Tabs in Nuxeo Platform Web UI are now ajaxified by default, providing a better user experience. This behaviour can be changed with the application property "uxeo.jsf.useAjaxTabs".

Path Displayed in Location Widget

The Location widget displays the full path so as to know which document exactly is selected, especially when two documents have the same title.

Like Operator on Policy Search

In Admin Center on Permissions tab, "Like" operator is used for searching on policy name field so as to be able to use "routing%" when searching, and then returning all the permissions set by the workflow engine.

Purging ACLs Archives Them

Purge action now moves the ACLs to "archive" state instead of being really deleted. This allows to easily audit who had access to what while

making sure that someone cannot access any content of the repository anymore.

For Developers

nuxeo-search-data element

A new element `nuxeo-search-data` has been added for being able to build Search Usages dashboards. This element makes use of the search events that are logged in the Nuxeo Platform Audit trail since Nuxeo Platform 7.4.

nuxeo-repository-data element

A new element, `nuxeo-repository-data` lets developers easily build custom Dataviz dashboards that explore content of the Nuxeo Platform repository, with aggregation and sort per type and properties of the documents stored in it.

Default Content Nuxeo Package

[A new Nuxeo Package](#) has been added so that when you start Nuxeo Platform the first time after having downloaded it, there is some demo content inside. You can consider this package as an exemple of how to bundle some content that can be easily installed on your Nuxeo Platform application.

Complex Properties and Default Values for Drag'n Drop

Metadata Screens

Drag'n Drop Collector screens now make use of a document model instead of a map. This makes it possible to handle complex properties as well as default values in the UI that pops out when doing a drag'n drop.

FileSystem Blob Provider

FileSystem blob provider has been added. It makes available as blobs specific files that are stored on a file system available from the server. Typical use case is when your content is handled by external applications that can't easily be updated for using Nuxeo REST API and for which it is easier to let the files on a standard file system. To use this feature, contribute a provider using:

```
<extension
target="org.nuxeo.ecm.core.blob.BlobManager"
point="configuration">
<blobprovider name="fs">
<class>org.nuxeo.ecm.core.blob.FilesystemBlobProvider</c
<property
name="root">/opt/nuxeo/nxserver/blobs</property>
</blobprovider>
</extension>
```

The configured root property above is mandatory and restricts under which path legal blobs will be allowed to be created. Then to create a blob pointing to a file under this root one can use:

```
BlobInfo blobInfo = new BlobInfo();
blobInfo.key =
"/opt/nuxeo/nxserver/blobs/foo/bar.pdf";
blobInfo.mimeType = "application/pdf";
BlobManager blobManager =
Framework.getService(BlobManager.class);
Blob blob = ((FilesystemBlobProvider)
blobManager.getBlobProvider("fs")).createBlob(blobInfo);
```

Internally the blob will then be stored in the database with a key of fs:foo/bar.pdf.

Implementing a New Binary Store Is Easier

All cloud binary managers like S3, Azure, Apache jclouds are now hosted in the same repository. They all inherit from a common code base so as to have only cloud client specific code in their modules. Also, we renamed all nuxeo-core-binarymanager-* to something more realistic like nuxeo-blob-*

CMIS 1.0 Can Be Used

Nuxeo Platform was already compatible with CMIS 1.1. A new endpoint has been added <http://localhost:8080/nuxeo/atom/cmis10>. It provides AtomPub CMIS 1.0 compatibility. This can be useful when using some connectors that do not support CMIS 1.1.

CMIS Also Handles File Download Redirect

When some URL download redirect logics are set in the blob manager, this is now taken into account by the CMIS service.

FS Exporter Addon

The addon fsexporter allows to export content out of Nuxeo Platform

with easy extensibility when a custom output format is required. This add-on is now part of the set of add-ons systematically released with Nuxeo Platform.

Remove Administrators Group From Suggestions

A new parameter has been added to the user suggestion widget: `hideAdminGroups`. It allows to make sure that admin group is not part of the suggested groups.

Batch Manager Uses the Transient Store

Now the Batch Manager relies on the Transient Store to allow several implementations among which the Redis one that is cluster aware.

Transient Store API

The Transient Store allows to store temporary blobs and associated parameters on a Nuxeo instance, outside the repository, thus the "transient" aspect. It is typically used by:

- The new [Batch Upload API](#) to store a batch of uploaded blobs until they are attached to a document.
- The `ConversionService` to store the `BlobHolder` resulting from an [Asynchronous Conversion Work](#).

The new [Transient Store](#) API allows this simply:

```
void putBlobs(String key, List blobs);  
List getBlobs(String key);  
void putParameters(String key, Map parameters);  
Map getParameters(String key);
```

Negative Caching for Directory Entries

Usually a read on a missing entry is rarely, if ever, attempted. But in the particular case of the `groupDirectory`, it happens a lot due to the fact that internally in Nuxeo user ids and group ids are often stored in the same field (for instance in ACLs). Therefore both the `groupDirectory` and the `userDirectory` have to be consulted before finding if a given id refers to a group or a user. With this improvement, to activate the negative caching, use a contribution like:

```
<require>...</require>
<extension
target="org.nuxeo.ecm.directory.sql.SQLDirectoryFactory"
point="directories">
<directory name="groupDirectory">
<negativeCaching>true</negativeCaching>
</directory>
</extension>
```

This improvement also adds two new metrics on directories:

- **nuxeo.directories.yourDirectoryName.cache.neghits:**
the number of hits on the negative cache, if activated.
- **nuxeo.directories.yourDirectoryName.cache.misses:**
the number of cache misses.

Renditions Can Be Stored by Default

A new boolean attribute **storeByDefault** has been added to the Rendition definition. This value is used for determining the default behaviour of the rendition service when a rendition is computed: store it or not. This is particularly useful when renditions are used through CMIS, as the CMIS api for rendition does not allow to specify the behaviour.

Per Rendition Type Last Modification Property Mapping

Property used to determine if a rendition needs to be re-computed is now configurable with a new attribute in the rendition definition: **sourceDocumentModificationDatePropertyName**. This is interesting in border cases of using renditions for instance when defining a rendition that is an export of the content of a folder. One would better rely on a date that is the last modification date of a given child document than the last modification date of the folder itself.

Prevent from Updating Files for a given Blob Provider

It is now possible to configure the list of blob providers for which the Web UI prevents the user from updating the file. Ways to update a file include:

- the file widget,
- the drag'n drop on the summary file widget,
- the drag'n drop in a folder (if the name is the same than the name of a blob stored in a child document, the file updates

the file property of that document),

- Nuxeo Drive synchronization.

To disable updates for a given blob provider, in the blob provider definition add the following property:

```
<property name="preventUserUpdate">true</property>
```

For instance you could have:

```
<extension
target="org.nuxeo.ecm.core.blob.BlobManager"
point="configuration">
<blobprovider name="googledrive">
<class>org.nuxeo.ecm.liveconnect.google.drive.GoogleDrive
<property name="preventUserUpdate">true</property>
</blobprovider>
</extension>
```

Renditions for Non Versionnable Documents Can Be Stored

Renditions can now be stored even for the ones associated with non-versionable documents, like folders. As an exemple, it is possible to store the zipTreeExport rendition.

New Parameters for Convert Adapter

Adapter for asynchronous conversions `@convert` now also handles type and format parameters. An API on the `ConversionService` has also been added to launch a conversion based on mime type (`#convertToMimeType()`).

Schema Update from Scalar to Array with MongoDB

When using MongoDB storage, it is now possible to change a schema from a scalar value to an array without any trouble.

NXQL Escaping with Automation

`Fn.nxqlEscape` can now be used in Automation MVEL expressions.
Ex:

```
SELECT * FROM Document WHERE ecm:path STARTSWITH
'@{Fn.nxqlEscape(Document.path)}'
```


Elasticsearch Query by Ancestor Id

ecm:ancestorId is now supported by Elasticsearch page providers.
As a reminder, **ecm:ancestorId** allows to find all documents who are rooted to a given document, provided its id.

Abandoned Addons

The following addons are abandoned:

- BIRT Addon: the new dataviz approach is recommended for analytics.
- JBPM: You should use Content Routing for all of your workflows on Nuxeo Platform LTS 2015.

For Administrators

Azure Object Store Binary Manager

A new binary manager allows to store binary content on Microsoft Azure Binary Storage. This connector is [available](#) as a Nuxeo Package on the Nuxeo Marketplace.

Server Don't Start on Pending Actions Failure

When some pending actions fail, the server now doesn't start, unless the option `--ignore-missing` is used.

```
$ echo "install fail" >
nxserver/data/installAfterRestart.log
$ ./bin/nuxeoctl start -q
org.nuxeo.connect.update.PackageException: Package not
found: fail
Pending actions execution failed. The commands file
has been moved to: /.../nuxeo-cap-7.10-SNAPSHOT-
tomcat/nxserver/data/installAfterRestart.log.bak
Start interrupted due to failure on pending actions.
You can resume with a new start; or you can restore
the file 'installAfterRestart.log', optionally using
the '--ignore-missing' option.

$ echo "install fail" >
nxserver/data/installAfterRestart.log
$ ./bin/nuxeoctl start -q --ignore-missing
Unable to install unknown package: fail
Server started with process ID 4957.
```

Studio Link and Project Name in Admin Center

The Studio link from Admin Center was redirecting you to the welcome page of Nuxeo Studio. It now redirects you to the exact Studio project against which you registered your Nuxeo Platform instance.

Furthermore user can see in the Update Center the name of the project the Nuxeo Platform instance is registered against.

Optimized LDAP Requests

LDAP requests have been optimised to fetch only necessary attributes so as to minimize the traffic.

Elasticsearch Audit and UID Sequence Indexes Are

Configurable

The Elasticsearch index names used for the UID sequencer and for audit logs are now configurable through the following properties in `nuxeo.conf`:

- Name of the Elasticsearch index for audit logs
`audit.elasticsearch.indexName=${elasticsearch.indexName}-audit`
- Name of the Elasticsearch index for the uid sequencer
`seqgen.elasticsearch.indexName=${elasticsearch.indexName}-uidgen`

As a reminder, the audit Elasticsearch index is mostly used by `ESAuditBackend` and the UID sequencer index by `ESUIDSequencer`. This allows to have several Nuxeo instances sharing the same Elasticsearch cluster.

We use the `elasticsearch.indexName`. Since the default value for `elasticsearch.indexName` is `nuxeo`, the resulting default values for the new properties are:

- `audit.elasticsearch.indexName=nuxeo-audit`
- `seqgen.elasticsearch.indexName=nuxeo-uidgen`

More Comprehensive Message on Restricted Package

Installation Error

`nuxeoctl mp-add` command provides a more comprehensive message when user try to install a Nuxeo package that requires to be registered.

Better Management of Marketplace Download Timeout

Some improvements have been implemented with package download:

- spinner progress bar
- no more limit on the whole download time
- per-download timeouts set on connection (30s) and socket read (120s)
- improved details on download error reporting

- global download interruption ("Packages download was interrupted") on first server error other than "not found", "access refused" or "registration required"

Quartz Library Is Upgraded

Quartz 2.2.2 is now used. Quartz is the scheduler component.
