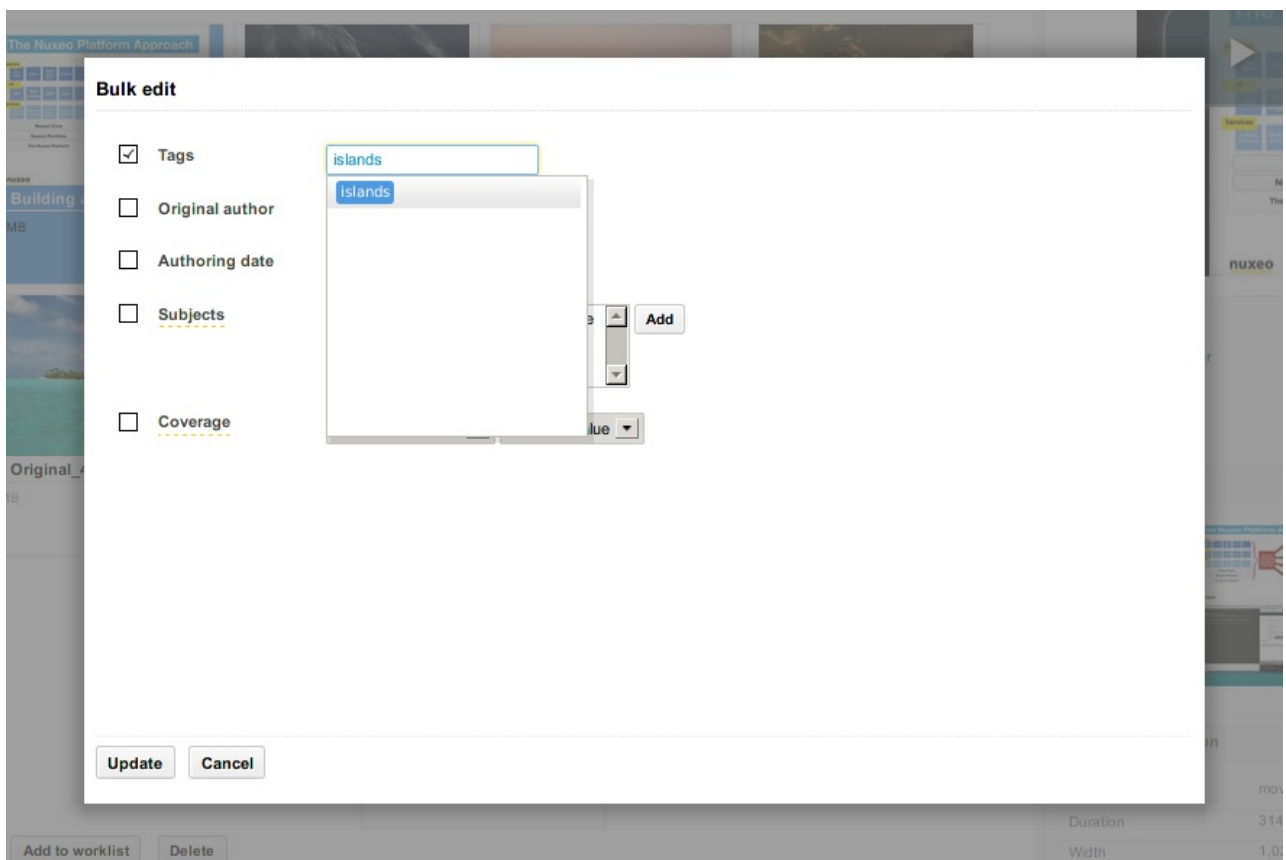


## For Users

### DAM

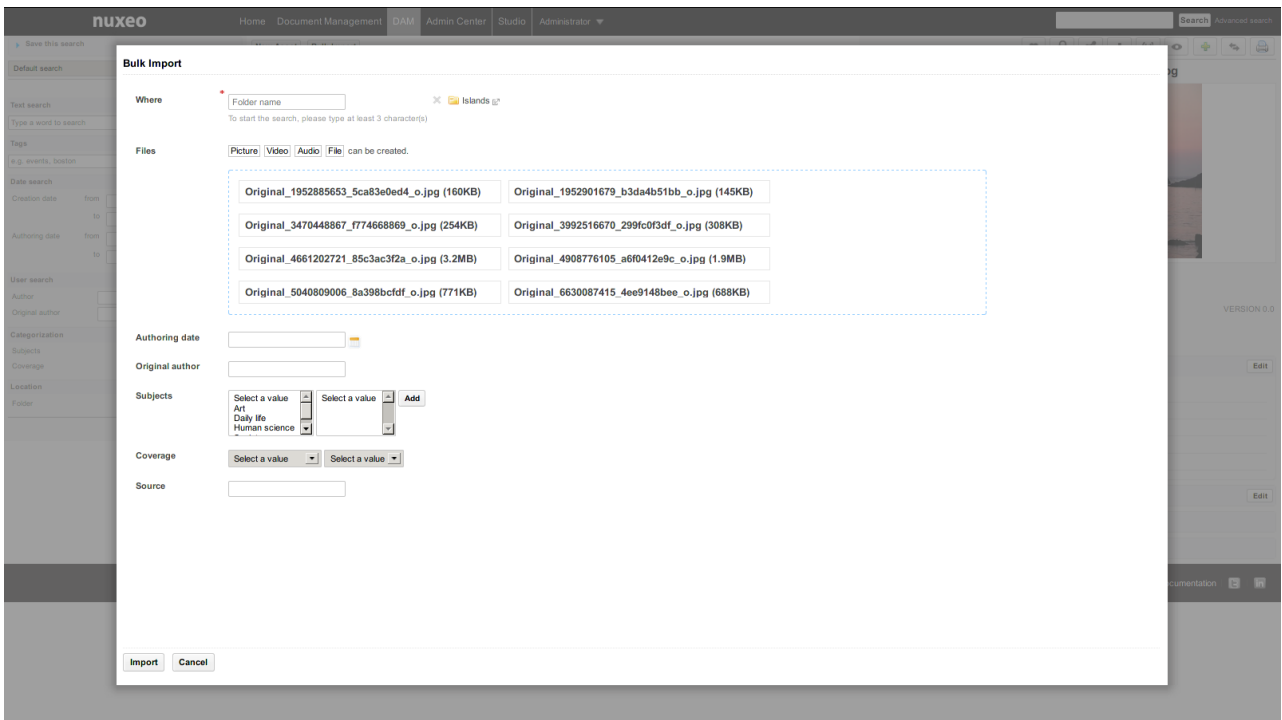
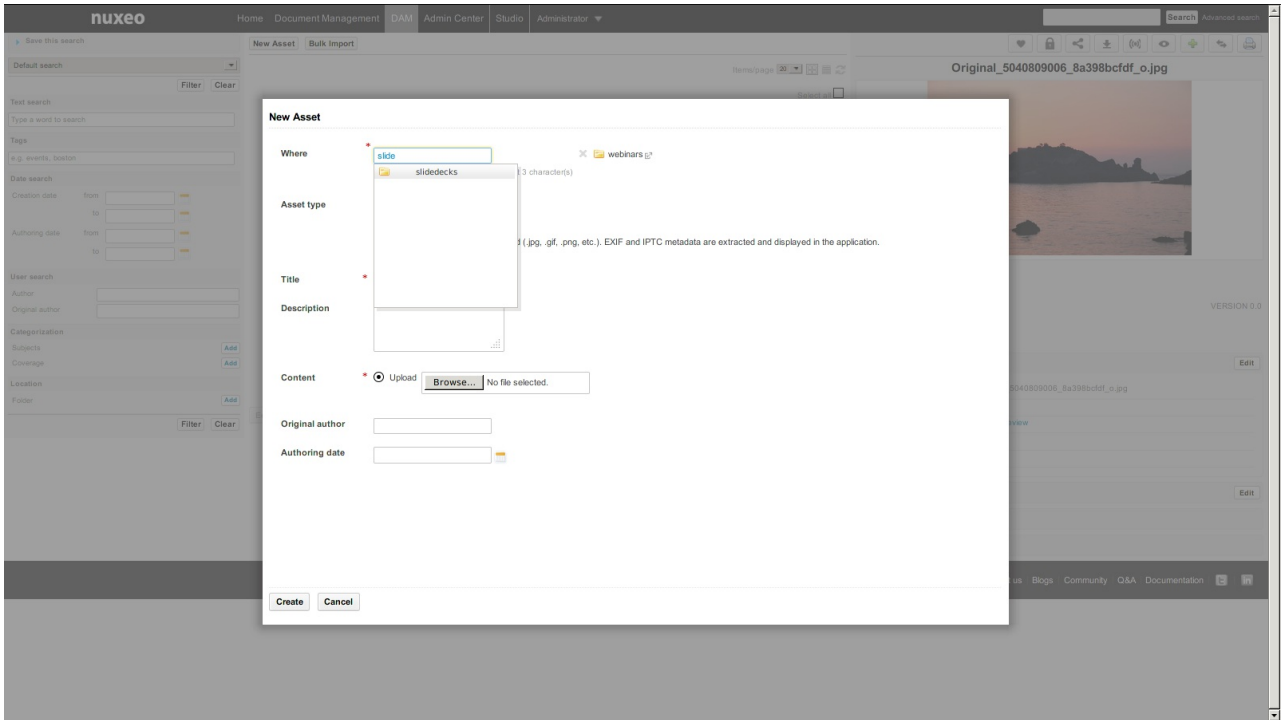
#### Bulk Tagging

In the bulk edit form, you can now add tags to the selected list of documents.



## Asset Creation

When creating a new asset or bulk importing assets, you can choose the location where to import them. Only the folders where the user has the create permission are selectable.



## Assets View

The view or edit layouts configured on the Document types (the ones you can configure on your own document type on Studio for instance) are displayed in the Metadata section.

## Single Asset View

Improve the single asset view of an asset (new layout accessible through the permalink).

The screenshot displays the Nuxeo Single Asset View for an asset titled "Nuxeo Island Rocks". The interface is divided into several sections:

- Header:** Includes the Nuxeo logo and navigation links: Home, Document Management, DAM, Admin Center, Studio, and Administrator.
- Image:** A large central image showing a tropical island with green trees and a blue sky, surrounded by clear turquoise water.
- Metadata:** A table listing key information:

Title	Nuxeo Island Rocks
Description	That's the description field.
Contact	<a href="#">Preview</a>
Original author	Pepper Anselmo
Authoring date	30/07/2013
- Intellectual Property:** A table listing rights and broadcast information:

Media type	Print
Geographical broadcast	United States of America
Copyright holder	Pepper Anselmo
Licence	All rights reserved
Embargo date	30/07/2013
Expiration date	30/07/2013
- EXIF:** A table listing technical camera data:

Date	13/09/2008
Width	2704
Height	1989
Orientation	TOP_LEFT
Equipment	Canon Canon PowerShot SD700 IS
FNumber	5.6
Exposure time	1/500
ISO	
- Interactions:** Below the image, there are buttons for "Add a tag", "Project", and social media icons (heart, share, comment).
- Footer:** Copyright © 2001-2013 Nuxeo and respective authors. Links for Contact us, Blogs, Community, Q&A, and Documentation.

## Misc Improvements

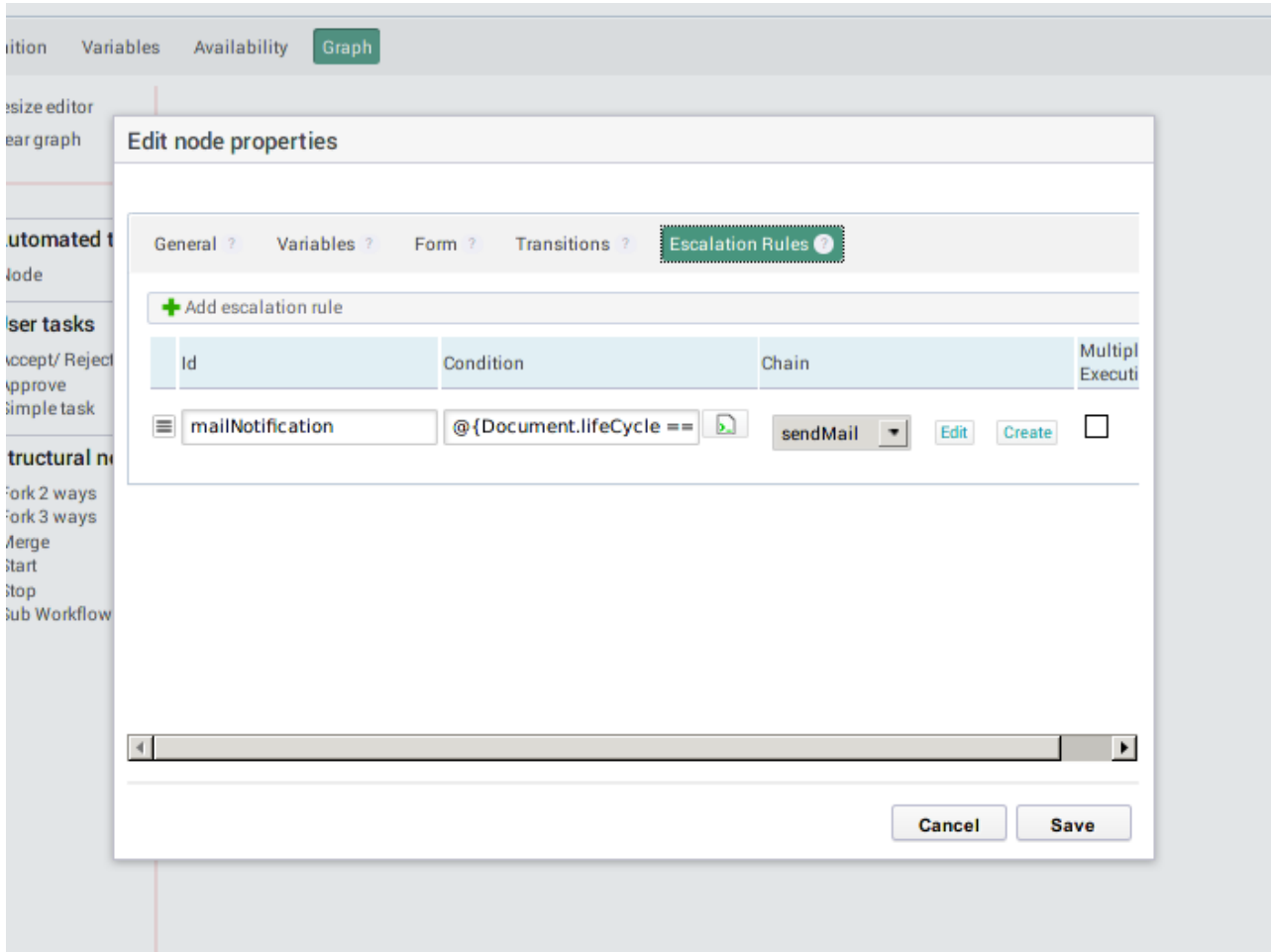
Now you can use the HTML5 DnD on the Bulk import form.

If your import fails for some reason (for instance, if you forgot to fill a required field), you get an appropriate error message and you don't have to select DnD files to import again.

## Workflow

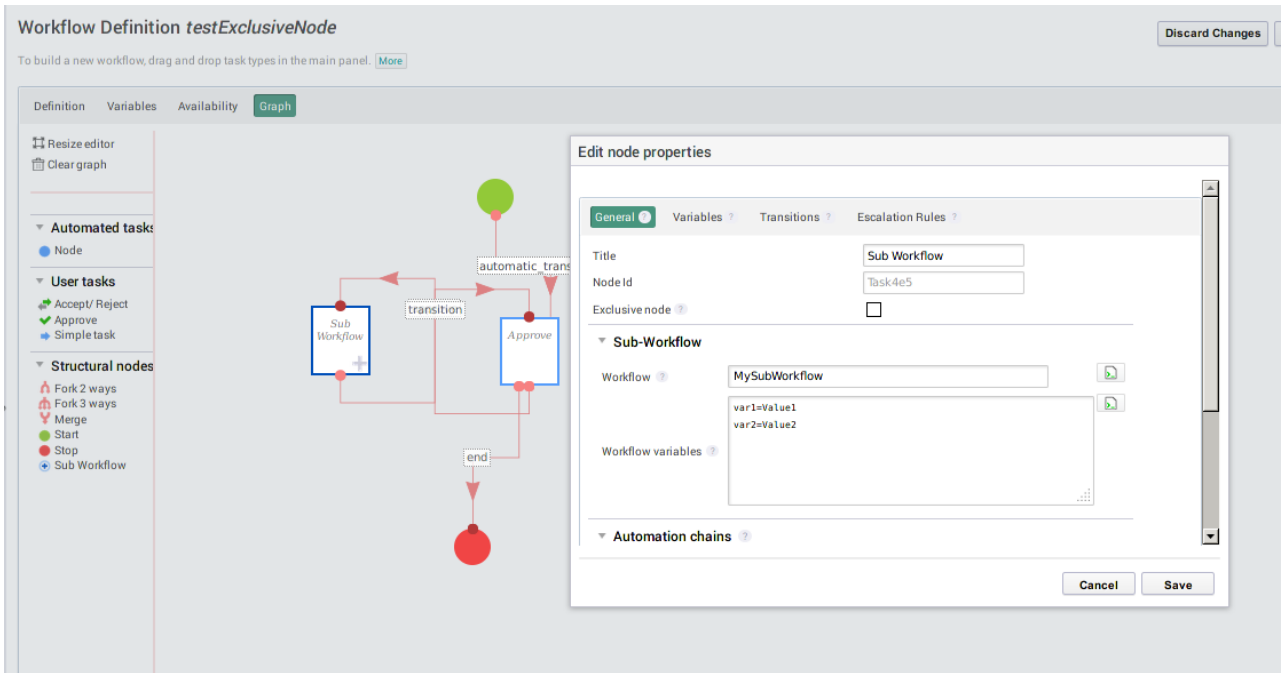
### Escalation Rules

Escalation rules let you run an operation chain when a condition is true. These conditions are evaluated every 5 minutes by a configurable scheduler. They can be executed one time (default behaviour) or multiple times.



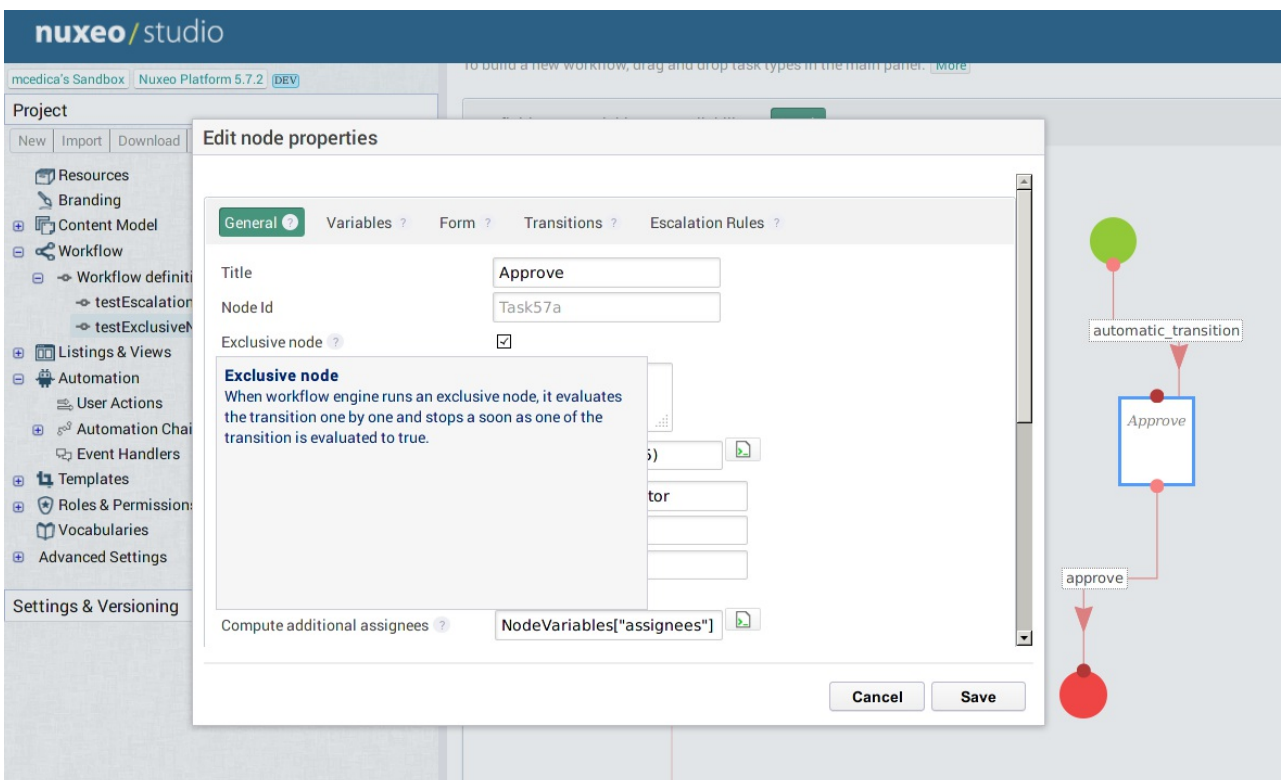
## Sub-Workflow

We have added a new node called SubWorkflow. As you have already guessed it, it lets you embed sub-workflow into existing one. You can also give them parameters, escalation rules, input and output chains, and transitions, like a regular node.



## Exclusive Node

When the workflow engine runs an exclusive node, it evaluates the transitions one by one and stops at the first one evaluated to true.

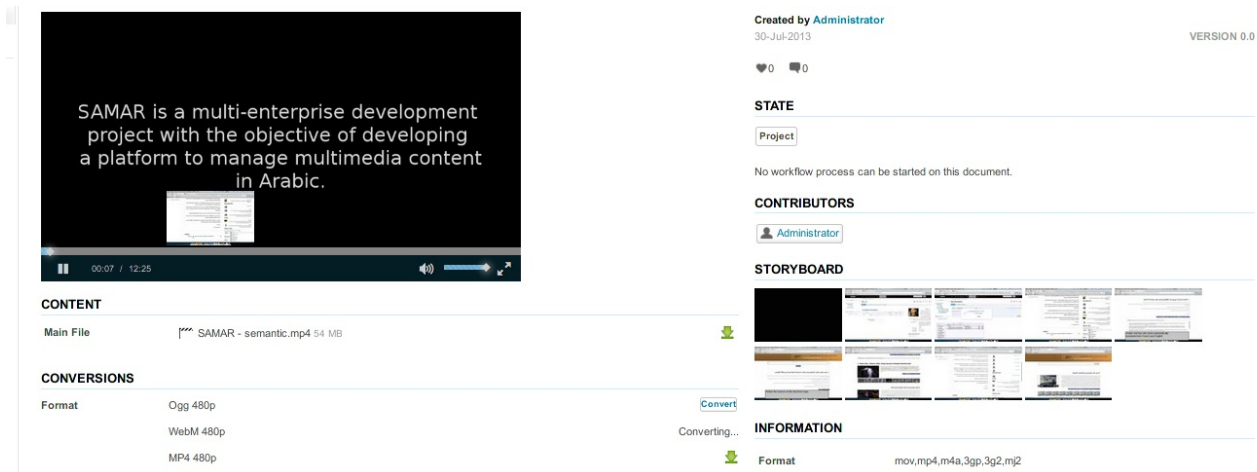


## New Workflow Operations

We've added new operations to cancel a workflow, resume a node, get the opened tasks, and complete a task and also improved the *Start Workflow* and *Set Workflow variable* operations.

## Video

The video.js player has been updated to version 4.1.0. This will enable us to benefit from their new plugin architecture. The first plugin we've integrated shows thumbnails of the video while hovering on the progress bar.



The screenshot displays a video player interface. The main video player shows a black screen with white text: "SAMAR is a multi-enterprise development project with the objective of developing a platform to manage multimedia content in Arabic." Below the video player, there are several sections:

- CONTENT**: Main File SAMAR - semantic.mp4 54 MB
- CONVERSIONS**: Format Ogg 480p, WebM 480p, MP4 480p
- STORYBOARD**: A grid of video thumbnails.
- INFORMATION**: Format mov,mp4,m4a,3gp,3g2,mj2

On the right side, there is a metadata panel:

- Created by Administrator
- 30-Jul-2013
- VERSION 0.0
- STATE: Project
- CONTRIBUTORS: Administrator

## Ergonomy and UX

Tables, forms and layout have been reviewed and simplified. A focus has been added to simplify the fill of inputs and make the document edition smoother.

We have started to update the icons throughout the platform. This effort will continue until the next LTS release.

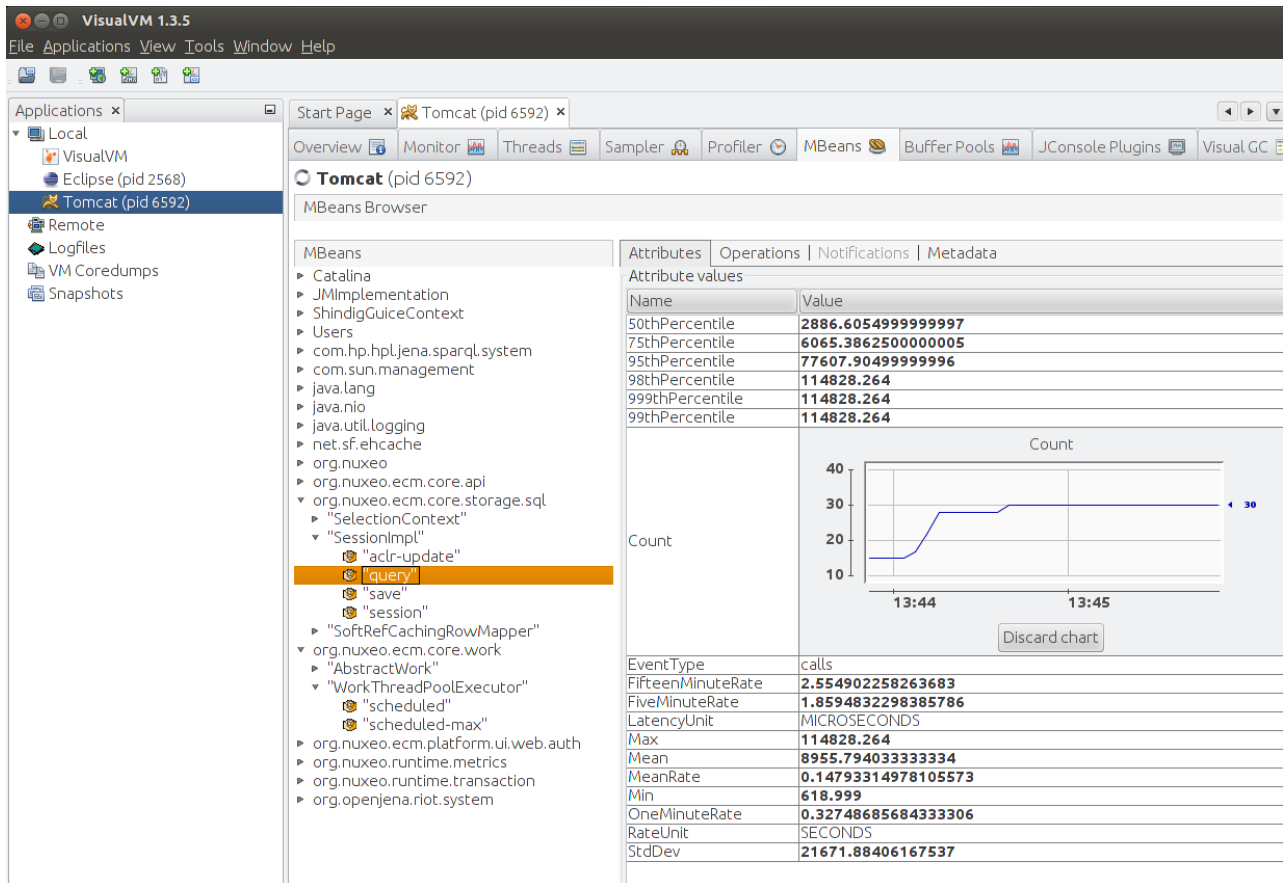
## Translations

Some contributions have been integrated to Nuxeo. Thanks to Patrick Turcotte for the French (Canada) translation and to Jordi Mallach for the Catalan translation.

# For Administrators

## Monitoring

Upgrade to the latest [metrics codahale lib](#) providing more metrics and fixes. Also note that metrics names have changed. Read more in our [documentation](#).



## Multi-Tabs Navigation

Nuxeo now handles multiple-tabs navigation without having a "cannot restore view" error. A new StateManager has been introduced in order to store ViewStates per conversation. In other words, when a user navigates to a new tab, a new conversation is started and its view states are saved separately. If the user comes back to an old tab, he will not have "view expired" exceptions anymore.

By default, 4 conversation states are saved following a LRU policy (Least Recently Used).

Each conversation state still has max 4 saved view states.

This means that potentially, a maximum of 16 view states are saved in session.

You can adapt these settings using the appropriate parameters in nuxeo.conf.

## Nuxeo.conf New Parameters

Parameter	Default Value	Description
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.notification.eMailSigner	The Nuxeo team	Signer of the sent e-mail.



## For Developers

### Content Automation

We made a lot of progress on the Content Automation API.

It's possible to contribute a chain into another chain -> Composite Automation Chain

It's possible to add parameters to a chain

Chains contributed are displayed into the Automation documentation with operations listing, parameters and input/output.

An Automation Client Test Suite in Java (or Java Test Compatibility Kit -> TCK) is provided for helping people understanding how Automation is working remotely. You can read our [documentation](#) for the details.

[Managing business objects](#) (Plain Old Java Object client side for mapping Nuxeo Document Model Adapter server side) is now available.

The goal is to manipulate business object and avoid direct DocumentModel manipulation on client side:

```
pull : BusinessObject (POJO) <----JSON---- DocumentModelAdapter <---- DocumentModel  
push : BusinessObject (POJO) -----JSON---> DocumentModelAdapter -----> DocumentModel
```

## Document Oriented REST API

We have a new [document oriented API](#) for CRUD operation.

### Create

```
POST /nuxeo/site/api/id/{idOfTheDoc}
{
  "entity-type": "document",
  "name": "newName",
  "type": "File",
  "properties": {
    "common:icon": "/icons/domain.gif",
    "common:icon-expanded": null,
    "common:size": null
  }
}
```

### Read

```
GET /nuxeo/site/api/path/{pathOfTheDoc}
GET /nuxeo/site/api/id/{idOfTheDoc}
```

### Update

```
PUT /nuxeo/site/api/id/{idOfTheDoc}
{
  "entity-type": "document",
  "repository": "default",
  "uid": "37b1502b-26ff-430f-9f20-4bd0d803191e",
  "properties": {
    "common:icon": "/icons/domain.gif",
    "common:icon-expanded": null,
    "common:size": null
  }
}
```

### Delete

```
DELETE /nuxeo/site/api/id/{idOfTheDoc}
```

### Domain model object and Automation

You can use [Business objects with Automation](#). They are also binded on the REST API and you just have to use the same semantics than for document resources.

## Updating a business object

```
PUT /nuxeo/site/api/path/{pathOfTheDoc}/@bo/BusinessBeanAdapter
{
  "entity-typee": "BusinessBeanAdapter"
  "value": {
    id: "37b1502b-26ff-430f-9f20-4bd0d803191e",
    "type": "Domain",
    "title": "Default domain"
    "description": "My new description"
  }
}
```

## Richfaces

Nuxeo uses a custom version of Richfaces based on 3.3.1 GA, This custom version of Richfaces is now available on [GitHub](#) with all Nuxeo's patches.

## CMIS

We've upgraded OpenCMIS to 0.9.0 . This gives us the new CMIS 1.1 browser bindings but we haven't finished implementing all the new features yet.

## Content Views

On Nuxeo side, filter can be shown unfolded by default, and some content view templates have been redefined to ease the customization.

On Studio side, the "filter unfolded" option is presented for 5.7.2, and events as well as the page provider class and properties can now be configured also.

## CORS support

If you do cross-domain requests from any JavaScript client to access WebEngine resources or Automation APIs, there's a chance that your browser forbids it. [CORS](#) allows you to communicate with Nuxeo from another domain using XMLHttpRequests.

We've added [a new extension point](#) based on Vladimir Dzhuvinov's universal CORS filter, which allows you to configure on which URL cross-origin headers are needed.

## Scan Importer

The scan importer has been improved. You can choose if you want to create the import document container each time you launch an import. You can also choose to update an existing document instead of creating a new one.

## New Operation

## Get a Directory Projection

**Category:** Services

**Operation Id:** [Directory.Projection](#)

Executes a query using given filter and return only the column \*columnName\*. The result is assigned to the context variable \*variableName\*. The filters are specified as key=value pairs separated by a new line. The key used for a filter is the column name of the directory. To specify multi-line values you can use a \\ character followed by a new line.

Example:

```
firstName=John  
lastName=doe
```

By default, the search filters use exact match. You can do a fulltext search on some specific columns using the fulltextFields. it's specified as comma separated columnName, for instance:

Example:

```
firstName,lastName
```

## Run For Each in new TX

**Category:** Execution Flow

**Operation Id:** [Context.RunOperationOnListInNewTx](#)

Run an operation in a new Transaction for each element from the list defined by the 'list' paramter. The 'list' parameter is pointing to context variable that represent the list which will be iterated. The 'item' parameter represent the name of the context variable which will point to the current element in the list at each iteration. You can use the 'isolate' parameter to specify whether or not the evaluation context is the same as the parent context or a copy of it. If the isolate is 'true' then a copy of the current contetx is used and so that modifications in this context will not affect the parent context. Any input is accepted. The input is returned back as output when operation terminate.

## Login Page

The login page can now be configured via an extension point, making it easier to upgrade when the login page is customized, and benefit from new features, like [OpenID authentication](#) provided by Nelson Silva.

Here is a sample configuration for the login page:

```
<extension target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService" po
<loginScreenConfig>
  <bodyBackgroundStyle>url("${org.nuxeo.ecm.contextPath}/img/my_login_background_img.png")
  <headerStyle></headerStyle>
  <footerStyle>display: none;</footerStyle>
  <loginBoxBackgroundStyle>url("${org.nuxeo.ecm.contextPath}/img/my_background_img.png") 0
  <loginBoxWidth>100px</loginBoxWidth>
  <logoUrl>${org.nuxeo.ecm.contextPath}/img/my_logo.png</logoUrl>
  <logoAlt>My logo Alt</logoAlt>
  <logoWidth>100px</logoWidth>
  <logoHeight>100px</logoHeight>
  <newsIframeUrl>https://www.nuxeo.com/embedded/dm-login</newsIframeUrl>
</loginScreenConfig>
</extension>
```

## Long Running Listener

You can now extend a new abstract class called **AbstractLongRunningListener**. Its purpose is to help building an Asynchronous listeners that will handle a long running process.

By default, PostCommitEventListener is executed in a Worker that will take care of starting/committing the transaction.

If the listener requires a long processing, this will create long transactions which should be avoided. To avoid this behavior, this base class split the processing in 3 steps:

Pre processing: transactional first step

Long running: long running processing that should not require transactional resources

Post processing: transactional final step

To manage sharing between the 3 steps, a simple Map is provided.

## Move To Tomcat 7

Tomcat has been upgraded to version 7.0.42. The main improvement is the upgrade to the Servlet 3.0 API. Take a look at the [details](#).

---