10.2 Upgrade Notes (Nuxeo Issue Tracker)

JQL Query: ((project = "NXP" AND fixVersion = "10.2") or (project=ELEMENTS AND fixVersion="2.3.2")) AND resolution in (Fixed, Done) AND ("Impact type" = "API change" OR "Upgrade notes" is not EMPTY) ORDER BY component ASC, key ASC
Sorted by: Component/s ascending, then Key ascending
1–29 of **29** as at: **26/07/18 13:52**

Key	Summary	Impact type	Upgrade notes
NXP-24933	Don't store annotation document in a placeless way		A new configuration property nuxeo.annotations.placeless.storage is available to chang the annotations are stored. If set to true (default), the annotations are placeless documents. If set to false, the annotations are stored in a hidden folder. This folder is created under domain of the annotated document, or under the root if no domain.
NXP-24480	Cannot run automation subchain with parameters	API change	In OperationContext: - Removed void push(Map<String, ?> map) - Removed void pop(Map<String, ?> map) - Changed Object push(String type, Object obj) to void push(String type, Object obj) - Removed boolean containsKey(Object key) - Added Map<String, Object> getChainParameters() - Added Map<String, Object> putChainParameters(Map<String, ?> parameters) - Added Map<String, Object> removeChainParameters() - Added Object getChainParameter(String key) - Added T callWithChainParameters(Callable<T> callable, Map<String, Object> param - Added T callWithContextVar(Callable<T> callable, String key, Object value)

NXP-24583	Make UserGroup.Suggestion operation handle fullname search		Since Nuxeo 10.2 and 9.10-HF03, the `SuggestUserEntries` operation performs a fu user search, e.g. typing "John Do" returns the user with first name "John" and last name
			In Nuxeo 9.10-HF03, the previous behavior (not returning any user when typing a full na can be re-activated by using
			`<extension target="org.nuxeo.runtime.ConfigurationService"`
			`point="configuration">`
			`  <property name="org.nuxeo.automation.user.suggest.fullname">false</proper`
			`</extension>`
			In any case, typing "John" still returns the "John Doe" user and possibly other users suc "John Foo". Respectively, typing "Do" returns the "John Doe" user and possibly other us such as "Jack Donald".

NXP-25020	Relax import constraints for Automation Scripting	Configuration Change	It's now possible to allow specific Java classes to be used via Automation Scripting, by we add:
			- java.util.ArrayList
			- java.util.Arrays
			- java.util.Collections
			- java.util.UUID
			- org.nuxeo.runtime.transaction.TransactionHelper
			- org.nuxeo.ecm.core.api.Blobs
			- org.nuxeo.ecm.core.api.impl.blob.StringBlob
			- org.nuxeo.ecm.core.api.impl.blob.JSONBlob
			Other classes can be added, and previously-allowed ones denied, through:
			`<require>org.nuxeo.automation.scripting.classfilter</require>`
			`<extension`
			`target="org.nuxeo.automation.scripting.internals.AutomationScriptingComponent`
			`point="classFilter">`
			`  <classFilter>`
			`    <allow>com.example.MyClass</allow>`
			`    <allow>com.example.MyOtherClass</allow>`
			`    <deny>org.nuxeo.runtime.transaction.TransactionHelper</deny>`
			`  </classFilter>`
			`</extension>`
			One can use `<deny>*</deny>` to disallow all previously-allowed classes.
			The default contribution now allows scripting code like:
			`function run(input, params) {`
			`    var uuid = java.util.UUID.randomUUID().toString();`
			`    return org.nuxeo.ecm.core.api.Blobs.createJSONBlob("{'uuid': \"" + uuid +`
			`"\"}");`
			`}`

Key	Summary	Impact type	Upgrade notes

NXP-25197	NXP-25097 Write tests for bulk io part		Some test classes have been moved under nuxeo-core-io test part. But nuxeo-core-io t module is a dependency of nuxeo-core-test (former location) to enforce backward comp
			
			`TransactionalFeature` has been moved from `nuxeo-core-test` to `nuxeo-runt` `test`. With this move, we no longer need `ContainerFeature`. This feature has been removed, you can get same behavior by annotating your test class with:
			
			`@RunWith(FeaturesRunner.class)`
			`@Features(TransactionalFeature.class)`
			`@TransactionalConfig(autoStart = false)`
			`public class MyTest { ... }`
			
			By that, we also introduce two new features: `CoreIOFeature` and `CoreBulkFeature` deploy needed components to use/test the feature.

NXP-22450	Allow bare document references without repository name		The following two `documentResolver` restrictions with `idOnly` and `pathOnly` are nc possible:
			
			`<xs:restriction base="xs:string" ref:resolver="documentResolver"`
			`ref:store="idOnly" />`
			
			`<xs:restriction base="xs:string" ref:resolver="documentResolver"`
			`ref:store="pathOnly" />`
			
			Their semantics is to store only the id or only the path, without any prefixed repository. \ fetching the constraint, the document of the given id or path is resolved in the same rep as the current document.

NXP-24850	Enable new TrashService		**API Changes**
			
			The new TrashService fires dedicated events `documentTrashed` and `documentUntr` (hold by TrashService interface) instead of `lifecycle_transition_event`. The doc model passed in the event won't be saved by Nuxeo at the end.
			
			Keeping Old Trash implementation
			
			The trash implementation has changed in 10.2. If you want to keep previous implement relying on life cycle state, add the following contribution:
			
			`  <require>org.nuxeo.ecm.core.trash.service.migrator</require>`
			`  <extension target="org.nuxeo.runtime.migration.MigrationService"`
			`point="configuration">`
			
			`    <migration id="trash-storage">`
			`      <defaultState>lifecycle</defaultState>`
			`    </migration>`
			
			`  </extension>`
			
			If you want to migrate trash state to the new property model (ecm:isTrashed), follow the migrations steps.
			
			Trash Migration
			
			To migrate trash states to the new property model:
			1. Follow the step from section Keeping old trash implementation.
			2. In the Nuxeo Platform's JSF UI, go to Admin > System Information > Migration, cl button Migrate trashed state from lifecycle to property and wait until migration is completed.
			3. Remove the contribution added at step 1.

NXP-25097	BAF: Implement first Bulk Processor SetProperty		Some test classes have been moved under nuxeo-core-io test part. But nuxeo-core-io t module is a dependency of nuxeo-core-test (former location) to enforce backward comp
			
			`TransactionalFeature` has been moved from `nuxeo-core-test` to `nuxeo-runt` `test`. With this move, we no longer need `ContainerFeature`. This feature has been removed, you can get same behavior by annotating your test class with:
			
			`@RunWith(FeaturesRunner.class)`
			`@Features(TransactionalFeature.class)`
			`@TransactionalConfig(autoStart = false)`
			`public class MyTest { ... }`
			
			By that, we also introduce two new features: `CoreIOFeature` and `CoreBulkFeature` deploy needed components to use/test the feature.

Key	Summary	Impact type	Upgrade notes
NXP-24102	Upgrade Elasticsearch from 5.6.3 to 6.3.0		The Elasticsearch cluster must be upgraded to 6.x, customized elasticsearch settings a mapping may require update, follow the Nuxeo 10.2 upgrade procedure.
NXP-25214	Switch to Elasticsearch RestClient by default		The `elasticsearch.addressList` option in `nuxeo.conf` needs to be updated to a elasticsearch in HTTP REST, the format is `http://elasticnode:9200` instead of `elasticnode:9300` used by tl Transport client protocol. If you want to stay on the Transport client protocol, you have to explicitly set `elasticsearch.client=TransportClient` in your `nuxeo.conf`.
NXP-23745	Add increment API to Key/Value Store	API change	New APIs are available on `KeyValueStore`. Optimized storage of `Long` values: - put(String key, Long value) - put(String key, Long value, long ttl) - getLong(String key) - getLongs(Collection<String> keys) Atomic increment: - addAndGet(String key, long delta)
NXP-24471	Directory entry is not formatted after adding to a data table		Use **selected-item(s)** property from the suggestion widget instead of the **value** one in c make it work
NXP-24683	Fix tasks list overflow		It was fixed for 10.2
NXP-25160	Fix bulk nuxeo-delete-documents-action availability		The bulk permanent delete action when browsing the trash pill is not available anymore can either use the empty trash action or navigate to the trashed document to permanen delete it from the document info bar.
NXP-25323	Move page.js back to routing.html	Configuration Change	If index.jsp page is overriden the page.js import should be removed.
NXP-24499	Nuxeo Vision: Remove the constraint for a predefined list of features		The features to use (and sent to the provider) are no more checked against a predifined This allows for using any new feature available without waiting for an update of the plug using such new feature, caller should use the `getNativeObject` method and handle t results based on the provider's documentation.
NXP-24185	Adding an alert on a proxy generates an error		Since ~~NXP-24185~~, we allow to subscribe notification on published documents. To do that, Nuxeo clears the notification subscriptions from version during check in of document.
NXP-25158	Add a way to do not compute page provider aggregation on demand		You can add the skipAggregates=true http header when invoking the search rest endpo skip any aggregation computation on a page provider query.
NXP-24754	S3 Direct Upload - Switch Temp Credentials to STS:GetFederationToken	API change	

NXP-16398	Review jsessionid management in URL system		Since 10.2, Tomcat is configured by default with the `COOKIE` session tracking mode.
			This prevents Tomcat from appending `jsessionid` to the URLs, for example a file dov URL.
			In 10.2 and later, to disable the `COOKIE` tracking mode and keep Tomcat's out of the bo configuration, set the following property in `nuxeo.conf`:
			`session.config.tracking.mode.cookie=false`
			In 9.10-HF12, 8.10-HF33 and 7.10-HF43, the previous behavior is kept.
			The new behavior can be enabled by setting
			`session.config.tracking.mode.cookie=true`
			Session tracking mode implications
			If the `COOKIE` mode is:
			- **enabled**: the `jsessionid` parameter will never be appended to the URLs. Yet, cook need to be enabled in the brower.
			- **disabled**: the `jsessionid` parameter might be appended to some URLs, for instanc sharing a document permalink to an anonymous user or when clearing the browser's cookies. Yet, cookies don't need to be enabled in the browser.

<table>
<tr><th>Key</th><th>Summary</th><th>Impact type</th><th>Upgrade notes</th></tr>
<tr><td>NXP-24682</td><td>duplicate values shown when only one field should</td><td></td><td>

The `item` parameter previously used by **user_group_display_template.xhtml** in an in way (not documented) is replaced by the `data` parameter to avoid any confusion with a existing variable named `item`, typically an iteration variable of a `<ui:repeat>` tag.

As stated by the widget template's documentation, the `data` parameter is "the field valu display in case `info.type` is empty".

It can be used this way:

```
<ui:include src="user_group_display_template.xhtml">
  <ui:param name="data" value="#{valueToDisplay}" />
</ui:include>
```

See for instance:

- single_user_suggestion_widget_template.xhtml
- user_suggestion_widget_template.xhtml

</td></tr>
<tr><td>NXP-24945</td><td>Sanitize tag name at creation time</td><td></td><td>

In addition to the characters ' (single quote), \ (backslash), % (precent) and space, that already forbidden in a tag, the character / (forward slash) is now also forbidden.

All these characters are removed at creation time and when we try to get a tag or updat

</td></tr>
<tr><td>NXP-25087</td><td>Upgrade Kafka from 1.0.0 to 1.1.0</td><td></td><td>In addition to Kafka 1.0.0 cluster we also support Kafka 1.1.0 cluster.</td></tr>
<tr><td>NXP-24493</td><td>Use Tomcat instead of old Jetty 6 for tests</td><td></td><td>

Various Jetty* test classes are renamed to ServletContainer*:

- `JettyFeature` -> `ServletContainerFeature`
- `JettyTransactionalFeature` -> `ServletContainerTransactionalFeatur`
- `@Jetty(port = ...)` -> `@ServletContainer(port = ...)`

Configuring the servlet container to add servlets, filters or listeners is now done through extension points. The syntax is based on the standard web.xml, with an additional `cont` attribute:

```
  <extension target="org.nuxeo.runtime.server" point="servlet">
    <servlet context="/">
      <servlet-name>My Servlet</servlet-name>
      <servlet-class>org.nuxeo.example.MyServlet</servlet-class>
      <url-pattern>/*</url-pattern>
      <init-param>
        <param-name>myparam</param-name>
        <param-value>myvalue</param-value>
      </init-param>
    </servlet>
  </extension>
```

```
  <extension target="org.nuxeo.runtime.server" point="filter">
    <filter context="/">
      <filter-name>My Filter</filter-name>
      <filter-class>org.nuxeo.example.MyFilter</filter-class>
      <filter-mapping>
        <url-pattern>/*</url-pattern>
        <dispatcher>REQUEST</dispatcher>
        <dispatcher>FORWARD</dispatcher>
      </filter-mapping>
      <init-param>
        <param-name>myparam</param-name>
        <param-value>myvalue</param-value>
      </init-param>
    </filter>
  </extension>
```

```
  <extension target="org.nuxeo.runtime.server" point="listener">
    <listener context="/">
      <listener-class>org.nuxeo.example.MyListener</listener-class>
    </listener>
  </extension>
```

</td></tr>
</table>

<table>
<tr><th>Key</th><th>Summary</th><th>Impact type</th><th>Upgrade notes</th></tr>
<tr><td>NXP-24847</td><td>Allow configuring the blobProvider used by a KeyValueBlobTransientStore</td><td>Configuration Change</td><td>A KeyValueBlobTransientStore can now specify the ids of the key/value store and blob
to use, instead of defaulting to the name of the transient store itself:
<pre><code><extension target="org.nuxeo.ecm.core.transientstore.TransientStorageComponen
point="store">
  <store name="mytransientstore"
class="org.nuxeo.ecm.core.transientstore.keyvalueblob.KeyValueBlobTransientSt
    <property name="keyValueStore">mykeyvaluestore</property>
    <property name="blobProvider">myblobprovider</property>
    ...
  </store>
</extension>
</code></pre></td></tr>
<tr><td>NXP-23744</td><td>UIDSequencer based on Key/Value Store</td><td>Configuration Change</td><td>A new <code>KeyValueStoreUIDSequencer</code> is available, to store sequences in a key/value
The store is the same for all sequencers, but they are using different keys, prefixed by t
sequencer name.
<p>It can be used by doing for example:</p>
<pre><code><extension target="org.nuxeo.ecm.core.uidgen.UIDGeneratorService"
point="sequencers">
  <sequencer name="uidgen"
class="org.nuxeo.ecm.core.uidgen.KeyValueStoreUIDSequencer" default="true" />
</extension>

<extension target="org.nuxeo.runtime.ConfigurationService" point="configurati
  <!-- this is an example, "sequence" is already the default -->
  <property name="nuxeo.uidseq.keyvaluestore.name">sequence</property>
</extension>
</code></pre>
<p>Assuming the following configuration to define a suitable key/value store, for example:</p>
<pre><code><extension target="org.nuxeo.runtime.kv.KeyValueService" point="configuratior
  <store name="sequence"
class="org.nuxeo.ecm.core.mongodb.kv.MongoDBKeyValueStore">
    <property name="collection">sequence</property>
  </store>
</extension>
</code></pre></td></tr>
<tr><td>ELEMENTS-619</td><td>Suggestion widgets should be able to be initialized through selected-item attribute</td><td></td><td>You can now initialize the suggestion widget with <b>selected-item(s)</b> which returns the se
item(s) instead of ids.</td></tr>
<tr><td>ELEMENTS-563</td><td>Add nuxeo-directory-checkbox (multiple value) and nuxeo-directory-radio-group (single value) widgets</td><td></td><td>2 new <b>nuxeo-directory-checkbox</b> (multiple) and <b>nuxeo-directory-radio-group</b> (single
widgets are available to present and select directory entrie(s)</td></tr>
<tr><td>ELEMENTS-621</td><td>Make it possible to sort aggregates by title</td><td></td><td>Aggregation widgets can now be sorted by the label with the <b>sort-by-label</b> boolean pro
i.e. `<nuxeo-checkbox-aggregation sort-by-label .../>`</td></tr>
</table>