

Nuxeo Issue Tracker

JQL Query: ((project=NXP AND fixVersion > 9.10 AND fixVersion <= 10.10) OR (project=ELEMENTS AND fixVersion=2.4.0)) 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

1-117 of 117 as at: 23/01/19 13:41

Key	Summary	Impact type	Upgrade notes
NXP-24933	Don't store annotation document in a placeless way		<p>A new configuration property <code>nuxeo.annotations.placeless.storage</code> is available to change how the annotations are stored.</p> <p>If set to true (default), the annotations are placeless documents.</p> <p>If set to false, the annotations are stored in a hidden folder. This folder is created under the domain of the annotated document, or under the root if no domain.</p>
NXP-25110	Allow enabling GWT annotations		<p>The GWT annotations are enabled when using the old JSF preview that can be enabled by setting the <code>nuxeo.old.jsf.preview</code> property to true.</p> <pre><extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.old.jsf.preview">true</property> </extension></pre>
NXP-26360	Deprecate AuditBulker implementation		<p><code>StreamAuditFeature</code> has been removed because <code>AuditFeature</code> now deploys audit with stream.</p> <p><code>AuditBulkerFeature</code> has been created to deploy audit with deprecated <code>AuditBulker</code>.</p>

Key	Summary	Impact type	Upgrade notes
NXP-26578	LogEntryJsonWriter doesn't handle complex in ExtendedInfo	Data Persistence Change	<p>Complex types in ExtendedInfo in LogEntry are now serialized correctly.</p> <p>This is a breaking change because we used to serialize them as their <code>String</code> representation and we now serialize them as JSON object.</p> <p>For instance, we changed:</p> <pre>{ "extended": { "array": "[123,456]", "map": "{key1=value1,key2=value2}" } }</pre> <p>to:</p> <pre>{ "extended": { "array": [123, 456], "map": { "key1": "value1", "key2": "value2" } } }</pre>
NXP-24734	Stateless Authentication based on a JSON Web Token (JWT)	Configuration Change	<p>There is a new Java API to acquire a JWT token to authorize a user:</p> <pre>JWTService service = Framework.getService(JWTService.class); String token = service.newBuilder().build();</pre> <p>The builder can also be used to add specific claims in the token (only <code>CLAIM_SUBJECT</code> is meaningful to the authenticator for now) and a TTL.</p> <p>The token should then be propagated and passed by a third-party service wishing to connect to Nuxeo using the <code>Authorization: Bearer <token></code> request header. As a compatibility fallback, the request query parameter <code>access_token</code> can also be used.</p> <p>IMPORTANT</p> <p>For the <code>JWTService</code> to work, all the cluster servers must be configured with a common shared secret in <code>nuxeo.conf</code>:</p> <pre>nuxeo.jwt.secret=...</pre>

Key	Summary	Impact type	Upgrade notes
NXP-25062	Expose create and update flags through contribution for UserMapperBasedResolver		<p>Two new parameters are now available when configuring SAML authentication plugin:</p> <ol style="list-style-type: none"> 1. userResolverCreatelfNeeded to create the user if it does not exist in the repository (default value is true) 2. userResolverUpdate to update the user if present in the repository (default is value true) <p>When set to true, both parameters require a user directory that is not read-only.</p>
NXP-25887	Allow PORTAL_AUTH to use a different digest than MD5	Configuration Change	<p>To change the digest algorithm used by PORTAL_AUTH, use a contribution like:</p> <pre> <require>org.nuxeo.ecm.platform.ui.web.auth.WebEngineConfig</require> <require>org.nuxeo.ecm.platform.login.Portal</require> <extension target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService" point="authenticators"> <authenticationPlugin name="PORTAL_AUTH"> <loginModulePlugin>Trusting_LM</loginModulePlugin> <parameters> <parameter name="secret">...</parameter> <!-- shared secret between the portal and Nuxeo server --> <parameter name="maxAge">...</parameter> <!-- in seconds --> <parameter name="digestAlgorithm">SHA-512</parameter> </parameters> </authenticationPlugin> </extension> </pre> <p>The algorithm used must be one of those described in https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#MessageDigest</p> <p>For compatibility with previous Nuxeo versions, the default is MD5.</p>
NXP-25975	Transform OAuth 1 pre-filter into a regular Authentication plugin	Configuration Change	<p>To activate OAuth 1, an <authenticationChain> must be defined to include <plugin>OAUTH1_AUTH</plugin>.</p>

Key	Summary	Impact type	Upgrade notes																																						
NXP-22232	Rename operations WebUI.* related to JSF	API change	<table border="1"> <thead> <tr> <th>Previous names</th> <th>New names</th> </tr> </thead> <tbody> <tr><td>WebUI.AddErrorMessage</td><td>Seam.AddErrorMessage</td></tr> <tr><td>WebUI.AddInfoMessage</td><td>Seam.AddInfoMessage</td></tr> <tr><td>WebUI.AddMessage</td><td>Seam.AddMessage</td></tr> <tr><td>WebUI.ClearClipboard</td><td>Seam.ClearClipboard</td></tr> <tr><td>WebUI.ClearSelectedDocuments</td><td>Seam.ClearSelectedDocuments</td></tr> <tr><td>WebUI.ClearWorklist</td><td>Seam.ClearWorklist</td></tr> <tr><td>WebUI.ShowCreateForm</td><td>Seam.CreateDocumentForm</td></tr> <tr><td>WebUI.DownloadFile</td><td>Seam.DownloadFile</td></tr> <tr><td>WebUI.NavigateTo</td><td>Seam.NavigateTo</td></tr> <tr><td>WebUI.AddToClipboard</td><td>Seam.AddToClipboard</td></tr> <tr><td>WebUI.PushDocumentToSeamContext</td><td>Seam.PushDocument</td></tr> <tr><td>WebUI.AddToWorklist</td><td>Seam.AddToWorklist</td></tr> <tr><td>WebUI.RaiseSeamEvents</td><td>Seam.RaiseEvents</td></tr> <tr><td>WebUI.Refresh</td><td>Seam.Refresh</td></tr> <tr><td>WebUI.SetJSFOutcome</td><td>Seam.SetOutcome</td></tr> <tr><td>WebUI.DestroySeamContext</td><td>Seam.DestroyContext</td></tr> <tr><td>WebUI.InitSeamContext</td><td>Seam.InitContext</td></tr> <tr><td>WebUI.RunOperationInSeam</td><td>Seam.RunOperation</td></tr> </tbody> </table>	Previous names	New names	WebUI.AddErrorMessage	Seam.AddErrorMessage	WebUI.AddInfoMessage	Seam.AddInfoMessage	WebUI.AddMessage	Seam.AddMessage	WebUI.ClearClipboard	Seam.ClearClipboard	WebUI.ClearSelectedDocuments	Seam.ClearSelectedDocuments	WebUI.ClearWorklist	Seam.ClearWorklist	WebUI.ShowCreateForm	Seam.CreateDocumentForm	WebUI.DownloadFile	Seam.DownloadFile	WebUI.NavigateTo	Seam.NavigateTo	WebUI.AddToClipboard	Seam.AddToClipboard	WebUI.PushDocumentToSeamContext	Seam.PushDocument	WebUI.AddToWorklist	Seam.AddToWorklist	WebUI.RaiseSeamEvents	Seam.RaiseEvents	WebUI.Refresh	Seam.Refresh	WebUI.SetJSFOutcome	Seam.SetOutcome	WebUI.DestroySeamContext	Seam.DestroyContext	WebUI.InitSeamContext	Seam.InitContext	WebUI.RunOperationInSeam	Seam.RunOperation
Previous names	New names																																								
WebUI.AddErrorMessage	Seam.AddErrorMessage																																								
WebUI.AddInfoMessage	Seam.AddInfoMessage																																								
WebUI.AddMessage	Seam.AddMessage																																								
WebUI.ClearClipboard	Seam.ClearClipboard																																								
WebUI.ClearSelectedDocuments	Seam.ClearSelectedDocuments																																								
WebUI.ClearWorklist	Seam.ClearWorklist																																								
WebUI.ShowCreateForm	Seam.CreateDocumentForm																																								
WebUI.DownloadFile	Seam.DownloadFile																																								
WebUI.NavigateTo	Seam.NavigateTo																																								
WebUI.AddToClipboard	Seam.AddToClipboard																																								
WebUI.PushDocumentToSeamContext	Seam.PushDocument																																								
WebUI.AddToWorklist	Seam.AddToWorklist																																								
WebUI.RaiseSeamEvents	Seam.RaiseEvents																																								
WebUI.Refresh	Seam.Refresh																																								
WebUI.SetJSFOutcome	Seam.SetOutcome																																								
WebUI.DestroySeamContext	Seam.DestroyContext																																								
WebUI.InitSeamContext	Seam.InitContext																																								
WebUI.RunOperationInSeam	Seam.RunOperation																																								
NXP-24281	Add an EmptyTrash operation		A new EmptyTrash operation allows to permanently delete a Folderish's trash content																																						
NXP-24282	Add Trash/Untrash operations and firstAccessibleAncestor Json Enricher		Two new operations TrashDocument and UntrashDocument and a firstAccessibleAncestor Json Enricher to get the closest document's ancestor used to redirect when deleting a document.																																						

Key	Summary	Impact type	Upgrade notes
NXP-24480	Cannot run automation subchain with parameters	API change	<p>In OperationContext:</p> <ul style="list-style-type: none">▪ 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> parameters)▪ Added T callWithContextVar(Callable<T> callable, String key, Object value)
NXP-24583	Make UserGroup.Suggestion operation handle fullname search		<p>Since Nuxeo 10.2 and 9.10-HF03, the <code>SuggestUserEntries</code> operation performs a full name user search, e.g. typing "John Do" returns the user with first name "John" and last name "Doe".</p> <p>In Nuxeo 9.10-HF03, the previous behavior (not returning any user when typing a full name) can be re-activated by using</p> <pre><extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="org.nuxeo.automation.user.suggest.fullname">false</property> </extension></pre> <p>In any case, typing "John" still returns the "John Doe" user and possibly other users such as "John Foo". Respectively, typing "Do" returns the "John Doe" user and possibly other users such as "Jack Donald".</p>

Key	Summary	Impact type	Upgrade notes
NXP-25020	Relax import constraints for Automation Scripting	Configuration Change	<p>It's now possible to allow specific Java classes to be used via Automation Scripting, by default we add:</p> <ul style="list-style-type: none">▪ 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 <p>Other classes can be added, and previously-allowed ones denied, through:</p> <pre><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></pre> <p>One can use <deny>*</deny> to disallow all previously-allowed classes.</p> <p>The default contribution now allows scripting code like:</p> <pre>function run(input, params) { var uuid = java.util.UUID.randomUUID().toString(); return org.nuxeo.ecm.core.api.Blobs.createJSONBlob("{\"uuid\": \"\" + uuid + "\"}"); }</pre>

Key	Summary	Impact type	Upgrade notes
NXP-25598	Do not remove attachments from Document when ignore Blob property in PUT requests		<p>Since 10.3, it is now possible to update blob(s) through the REST API when PUTing a document. Nuxeo reads the JSON object for a blob property and tries to resolve a blob from the download url set in the data field. The format of the JSON object is:</p> <pre>{ "data": "http://SERVER/nuxeo/nxfile/repository/docid/xpath" }</pre> <p>All other fields are ignored.</p> <p>Nuxeo does the following when trying to resolve a blob:</p> <ul style="list-style-type: none">▪ fetch the document▪ fetch the referenced blob property▪ check the download permissions <p>If one of the above fails (null, not found, no permission, ...), you get back a Bad Request response (400) when saving the document.</p> <p>On 9.10-HF19, it is not enabled by default to avoid any behavior changes. To enable it, you need to contribute the following configuration property:</p> <pre><require>org.nuxeo.ecm.core.automation.core.properties</require> <extension point="configuration" target="org.nuxeo.runtime.ConfigurationService"> <property name="nuxeo.automation.object.blob.decoder.enabled">true</property> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-25075	Add new Nuxeo AWS service to get credentials and other config	API change, Configuration Change	<p>A new template aws is available, to define the AWS configuration. When activated (which is automatically done by the marketplace-amazon-s3 package), the following nuxeo.conf properties are available:</p> <ul style="list-style-type: none"> nuxeo.aws.accessKeyId nuxeo.aws.secretKey nuxeo.aws.region <p>They are optional, and if not present the default AWS SDK mechanism for configuring them will be used (environment variables, Java system properties, local AWS profile, container-specific configuration (ECS/EC2)). See https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/credentials.html#credentials-default and https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/java-dg-region-selection.html#default-region-provider-chain for more information.</p> <p>Instead of using the template, the service can also be configured manually:</p> <pre><extension target="org.nuxeo.runtime.aws.AWSConfigurationService" point="configuration"> <configuration> <accessKeyId>MY_ACCESS_KEY_ID</accessKeyId> <secretKey>MY_SECRET_KEY</secretKey> <region>MY_REGION</region> </configuration> </extension></pre> <p>From Java there are two new APIs to get this information:</p> <ul style="list-style-type: none"> NuxeoAWSCredentialsProvider.getInstance() NuxeoAWSRegionProvider.getInstance().getRegion() <p>These credentials and region providers will default to standard AWS SDK behavior if no Nuxeo configuration is provided (see above).</p>
NXP-24421	Allow saving in documents blobs coming from a non-document blob provider		<p>A blob provider use by the transient store has to be configured with <property name="transient">true</property> for the blobs to be properly copied when saved to other blob providers.</p>
NXP-25525	Expose the "pathStyleAccess" s3 client option within a new configuration key	Configuration Change	<p>To configure path-style access, use the configuration property:</p> <pre>nuxeo.s3storage.pathstyleaccess=true</pre> <p>Or if configuring from an extension point:</p> <pre><extension target="org.nuxeo.ecm.core.blob.BlobManager" point="configuration"> <blobprovider name="default"> <class>org.nuxeo.ecm.core.storage.sql.S3BinaryManager</class> <property name="pathstyleaccess">true</property> ... </blobprovider> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-25540	Use more secure encryption algorithm for AESBinaryManager	Data Persistence Change	<p>Before Nuxeo 10.3, the encryption used was based on AES/CBC/PKCS5Padding which has been found to be insecure (susceptible to padding oracle attacks).</p> <p>Starting with Nuxeo 10.3, a more secure encryption algorithm, AES/GCM/NoPadding, is used. If you're migrating data from an earlier version and therefore still need to use the old insecure encryption algorithm, you must add the option <code>useInsecureCipher=true</code> to the <code>nuxeo.core.binarymanager_key</code> in <code>nuxeo.conf</code>.</p> <p>See https://doc.nuxeo.com/nxdoc/implementing-encryption/#filesystem-aes-encryption for details.</p>
NXP-26581	Different Transient Stores must use different namespaces	Configuration Change	<p>Transient stores with different names now use different storage.</p> <p>To configure transient stores you can now just configure the default one. If an internal Nuxeo service requests a specific non-default one, its configuration will take into account the default configuration in addition to whatever (if anything) is configured for the specific one.</p> <p>For instance the default Nuxeo template now contains:</p> <pre><extension target="org.nuxeo.ecm.core.transientstore.TransientStorageComponent" point="store"> <store name="default" class="org.nuxeo.ecm.core.transientstore.keyvalueblob.KeyValueBlobTransientStore"> <targetMaxSizeMB>-1</targetMaxSizeMB> <absoluteMaxSizeMB>-1</absoluteMaxSizeMB> <firstLevelTTL>240</firstLevelTTL> <secondLevelTTL>10</secondLevelTTL> </store> <store name="authorizationRequestStore"> <firstLevelTTL>10</firstLevelTTL> <secondLevelTTL>0</secondLevelTTL> </store> </extension></pre> <p>This shows that the implementation class needs to be defined only once in the default configuration, and other configurations can override some parameters if needed.</p> <p>As a further example, when the <code>redis</code> template is enabled and you explicitly set <code>nuxeo.transientstore.provider=redis</code> (which is <i>not</i> the default), then the following is added automatically and is enough to switch all transient stores to the new class (unless a specific non-default transient store has defined its own class):</p> <pre><extension target="org.nuxeo.ecm.core.transientstore.TransientStorageComponent" point="store"> <store name="default" class="org.nuxeo.ecm.core.redis.contribs.RedisTransientStore"/> </extension></pre> <p>In addition, if a <code>KeyValueBlobTransientStore</code> is configured without an explicit <code><keyValueStore></code> or <code><blobProvider></code>, it will automatically use a key/value store or blob provider named <code>transient_</code> followed by the transient store id.</p>

Key	Summary	Impact type	Upgrade notes
NXP-26594	Namespaced blob providers	API change, Configuration Change	<p>It's now possible to request a blob provider where there hasn't been any explicit configuration for it, using <code>BlobManager.getBlobProviderWithNamespace(id)</code>. If there is no blob provider configured for this id, then the default one will be used, although using a <i>namespace</i> based on the id to avoid collisions.</p> <p>For the filesystem-based blob provider, if the default is stored in <code>.../data/binaries/...</code> then a namespaced provider for <code>myid</code> will be stored in <code>.../data/binaries_myid/...</code></p> <p>For the S3 blob provider, the binaries of a namespaced provider <code>myid</code> will be stored in a S3 "subfolder" named <code>myid/</code> under the storage for default. Azure works similarly.</p> <p>Also, a blob provider whose name starts with <code>transient</code> will automatically be flagged as transient (see NXP-24424) to avoid manual configuration.</p>
NXP-25097	BAF: Implement first Bulk Processor SetProperty		<p>Some test classes have been moved under <code>nuxeo-core-io</code> test part. But <code>nuxeo-core-io</code> test module is a dependency of <code>nuxeo-core-test</code> (former location) to enforce backward compatibility.</p> <p><code>TransactionalFeature</code> has been moved from <code>nuxeo-core-test</code> to <code>nuxeo-runtime-test</code>. With this move, we no longer need <code>ContainerFeature</code>. This feature has been removed, you can get same behavior by annotating your test class with:</p> <pre>@RunWith(FeaturesRunner.class) @Features(TransactionalFeature.class) @TransactionalConfig(autoStart = false) public class MyTest { ... }</pre> <p>By that, we also introduce two new features: <code>CoreIOFeature</code> and <code>CoreBulkFeature</code> which deploy needed components to use/test the feature.</p>
NXP-25197	NXP-25097 / Write tests for bulk io part		<p>Some test classes have been moved under <code>nuxeo-core-io</code> test part. But <code>nuxeo-core-io</code> test module is a dependency of <code>nuxeo-core-test</code> (former location) to enforce backward compatibility.</p> <p><code>TransactionalFeature</code> has been moved from <code>nuxeo-core-test</code> to <code>nuxeo-runtime-test</code>. With this move, we no longer need <code>ContainerFeature</code>. This feature has been removed, you can get same behavior by annotating your test class with:</p> <pre>@RunWith(FeaturesRunner.class) @Features(TransactionalFeature.class) @TransactionalConfig(autoStart = false) public class MyTest { ... }</pre> <p>By that, we also introduce two new features: <code>CoreIOFeature</code> and <code>CoreBulkFeature</code> which deploy needed components to use/test the feature.</p>

Key	Summary	Impact type	Upgrade notes
NXP-26559	Use <code>keyStream(prefix)</code> in bulk action framework	Data Persistence Change	<p>The way we store contextual information for BAF has changed. Instead of having <code>BULK_ID:status</code> we now have <code>status:BULK_ID</code> (for instance).</p> <p>This makes all command submitted before upgrade not compatible with the Bulk Action Framework after the upgrade.</p> <p>There's no impact on finished command except that status can't be retrieved after the upgrade, but acceptable as it's transient data.</p> <p>There's impact on running command because after upgrading Nuxeo to 10.10, BAF won't be able to retrieve contextual information to finish submitted command before upgrade. As it we advise to upgrade Nuxeo from 10.3 to 10.10 after all BAF processing has finished. Otherwise a manual intervention on the <code>keyValueStore</code> is possible to reverse these key/value:</p> <ul style="list-style-type: none"> ▪ <code>BULK_ID:status</code> by <code>status:BULK_ID</code> ▪ <code>BULK_ID:command</code> by <code>command:BULK_ID</code>
NXP-24093	Replace use of <code>json-lib</code> with <code>jackson</code>		<p>Use <code>com.fasterxml.jackson.core:jackson-*</code> instead of <code>net.sf.json-lib:json-lib</code>.</p> <p>There's several way to benefit jackson from Nuxeo.</p> <ul style="list-style-type: none"> ▪ create a pojo and serialize it with <code>ObjectMapper</code> from Jackson ▪ create Nuxeomarshallers to be able to use it on REST API (every contributedmarshallers are automatically bound) or in a plain code with <code>MarshallerHelper</code>
NXP-24143	Clean up bugs/code smells reported by SonarQube on <code>nuxeo-drive-server</code>	API change	<p>Removed as deprecated since 5.6:</p> <pre>FileSystemItemManager#getTopLevelChildren(NuxeoPrincipal principal)</pre> <p>Stopped making Serializable uselessly:</p> <pre>FileSystemItem ScrollFileSystemItemList FileSystemChangeFinder FileSystemChangeSummary FileSystemItemChange SynchronizationRoots ActiveTopLevelFolderItemFactoryDescriptor ChangeFinderDescriptor FileSystemItemFactoryWrapper TopLevelFolderItemFactoryDescriptor NuxeoDriveActions</pre>

Key	Summary	Impact type	Upgrade notes
NXP-25910	Use NuxeoPrincipal instead of Principal in most APIs	API change	<p>The general upgrade rule is the Nuxeo code should not deal with a <code>java.security.Principal</code> anymore (except for internal authentication-related classes).</p> <h3>API changes</h3> <p><code>NuxeoPrincipal</code> is now used instead of <code>Principal</code> in the methods and constructors for these classes:</p> <ul style="list-style-type: none">▪ <code>CoreSession</code>▪ <code>DocumentEventContext</code>▪ <code>EventContext</code>▪ <code>EventContextImpl</code>▪ <code>InlineEventContext</code>▪ <code>OperationContext</code>▪ <code>QueryFilter</code>▪ <code>SQLQuery.Transformer</code>▪ <code>SecurityPolicy</code>▪ <code>SecurityPolicy.QueryTransformer</code>▪ <code>SecurityService</code>▪ <code>TrashService</code>▪ <code>UnboundEventContext</code>▪ <code>UserWorkspaceService</code>▪ <code>WebContext</code> <h3>Removed APIs</h3> <p>These APIs were previously deprecated or internal and have been removed.</p> <ul style="list-style-type: none">▪ <code>CoreInstance.getInstance()</code>▪ <code>CoreInstance.getNumberOfSessions()</code>▪ <code>CoreInstance.getRegistrationInfos()</code>▪ <code>CoreInstance.openCoreSession(repositoryName, Map context)</code>▪ <code>CoreInstance.openCoreSession(repositoryName, Principal principal)</code>▪ <code>DetachedNuxeoPrincipal</code>▪ <code>SimplePrincipal</code> (use <code>UserPrincipal</code> instead)

Key	Summary	Impact type	Upgrade notes
NXP-25543	Relocate nuxeo-platform-collections in nuxeo-service		<ul style="list-style-type: none">Collection Core types (Collection, Favorites, etc.), schema and facets have been merged in CoreExtensions.xmlnuxeo-platform-collections-jsf has disappeared. Resources and classes have been dispatched in nuxeo-jsf/nuxeo-platform-webapp-base and nuxeo-dm/nuxeo-platform-webapp-typesAll Collection-related automation operations have been moved to nuxeo-automation-featuresCollection and Favorites services have been kept but moved under nuxeo-services/nuxeo-collectionsUserWorkspaces service now contributes where should be located User's collections and Favorites by implementing CollectionLocationService.javaYou can use CollectionFeature.java to test a feature relying on Collections, need to add the following pom dependency: <pre><dependency> <groupId>org.nuxeo.ecm.platform</groupId> <artifactId>nuxeo-platform-collections-core</artifactId> <type>test-jar</type> <scope>test</scope> </dependency></pre>
NXP-25429	Handle Comment migration		<p>When upgrading from a previous Nuxeo version, if you want to keep (and later migrate) old comments, add the contribution:</p> <pre><require>org.nuxeo.ecm.platform.comment.manager.migrator</require> <extension target="org.nuxeo.runtime.migration.MigrationService" point="configuration"> <migration id="comment-storage"> <defaultState>relation</defaultState> </migration> </extension></pre>
NXP-26038	Make ConfigurationService.getPropertiesAsJson() support nested properties	API change	

Key	Summary	Impact type	Upgrade notes
NXP-25840	DWG file upload generates 500 error		<p>Since 9.10-HF22</p> <p>A new configuration property <code>nuxeo.preview.allowGenericHTMLConverter</code> is available to disable the generic HTML converter used by the preview when no converter is found to generate HTML. Disable it if you run into an issue when previewing formats that LibreOffice can't handle. It defaults to <code>true</code> to keep backward compatibility.</p> <p>To disable the generic HTML converter, use the following contribution:</p> <pre><require>org.nuxeo.ecm.platform.preview.properties</require> <extension point="configuration" target="org.nuxeo.runtime.ConfigurationService"> <property name="nuxeo.preview.allowGenericHTMLConverter">false</property> </extension></pre> <p>Since 10.3</p> <p>When calling a converter directly through its name, a check is done on its source MIME type to see if it can handle the input blob. A <code>ConversionException</code> is thrown if the converter can't handle it.</p> <p>A new configuration property <code>nuxeo.convert.enforceSourceMimeTypeCheck</code> is available to disable this behavior which defaults to <code>true</code>.</p> <p>To disable the check, use the following contribution:</p> <pre><require>org.nuxeo.ecm.core.convert.configuration</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.convert.enforceSourceMimeTypeCheck">false</property> </extension></pre>
NXP-22450	Allow bare document references without repository name		<p>The following two <code>documentResolver</code> restrictions with <code>idOnly</code> and <code>pathOnly</code> are now possible:</p> <pre><xs:restriction base="xs:string" ref:resolver="documentResolver" ref:store="idOnly" /></pre> <pre><xs:restriction base="xs:string" ref:resolver="documentResolver" ref:store="pathOnly" /></pre> <p>Their semantics is to store only the id or only the path, without any prefixed repository. When fetching the constraint, the document of the given id or path is resolved in the same repository as the current document.</p>
NXP-24044	Remove internal prefetch at the DocumentModel level	API change	A scalar list defined by a <code>xs:complexType</code> with just one element is now consistently represented as a <code>List</code> in Java, never an array.
NXP-24045	Remove deprecated <code>ScopedMap</code> and <code>ScopeType</code>	API change	Any code still using <code>ScopedMap</code> must change to <code>Map<String, Serializable></code> instead.

Key	Summary	Impact type	Upgrade notes
NXP-24089	Introduce separate CloseableCoreSession interface to close session	API change	<p>The various <code>CoreInstance.openCoreSession(...)</code> methods now return a <code>CloseableCoreSession</code>, which has a <code>close()</code> method and can be use with the <code>try-with-resources</code> pattern.</p> <p><code>CoreSession.close()</code> is deprecated and should not be used anymore.</p> <p>Any code previously calling <code>CoreSession.close()</code> has to be migrated to properly refer to the original <code>CloseableCoreSession</code> that was acquired, or, if the code flow is too complex, use a cast <code>((CloseableCoreSession) session).close()</code>.</p>
NXP-24725	Annotation type should extend Comment type		<p>A new facet <code>ExternalEntity</code> has been created to handle Annotations or Comments created from an external service like <code>ARender</code>.</p> <p>This facet allows to store the serialized entity.</p>
NXP-24850	Enable new TrashService		<p>API Changes</p> <p>The new <code>TrashService</code> fires dedicated events <code>documentTrashed</code> and <code>documentUntrashed</code> (hold by <code>TrashService</code> interface) instead of <code>lifecycle_transition_event</code>. The document model passed in the event won't be saved by Nuxeo at the end.</p> <p>Keeping Old Trash implementation</p> <p>The trash implementation has changed in 10.2. If you want to keep previous implementation relying on life cycle state, add the following contribution:</p> <pre><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></pre> <p>If you want to migrate trash state to the new property model (<code>ecm:isTrashed</code>), follow the Trash migrations steps.</p> <p>Trash Migration</p> <p>To migrate trash states to the new property model:</p> <ol style="list-style-type: none"> 1. Follow the step from section Keeping old trash implementation. 2. In the Nuxeo Platform's JSF UI, go to Admin > System Information > Migration, click the button Migrate trashed state from lifecycle to property and wait until migration is completed. 3. Remove the contribution added at step 1. <p>INFO: during migration, document in state <code>deleted</code> will receive the <code>isTrashed</code> property set to true but migrator will leave document in <code>deleted</code> state</p>

Key	Summary	Impact type	Upgrade notes
NXP-25279	Make the raw binary text available for processing	API change	<p>The <code>binaryTextUpdated</code> event now contains two properties of interest to know what was update exactly:</p> <ul style="list-style-type: none"> ▪ <code>systemProperty</code> contains the name of the property updated ▪ <code>systemPropertyValue</code> contains the value
NXP-25716	Simplify fulltext extraction	Configuration Change	<p>Fulltext maximum size is now 128 KB by default. To change this, the repository configuration can be updated to use another <code>fieldSizeLimit</code> value:</p> <pre><fulltext ... fieldSizeLimit="1048576"> ... </fulltext></pre> <p>A value of 0 means no limit.</p> <p>See https://doc.nuxeo.com/nxdoc/repository-configuration/#full-text for more</p>
NXP-25761	Fix Trash migration		<p><code>org.nuxeo.isTrashed.from.deleteTransition</code> introduced in 10.2 has been removed. Now, if you have migrated your Nuxeo instance to use new dedicated <code>isTrashed</code> property, calls to <code>CoreSession#followTransition</code> with <code>delete/undelete</code> transition are forwarded to <code>TrashService</code> without following the transition.</p> <p>Also, these transitions are deprecated as we DO NOT follow them anymore. As a consequence, after migration or on a fresh instance, documents can't have <code>deleted</code> state anymore.</p> <p>As a consequence of this backward mechanism, following these transitions on proxy will remove them.</p>
NXP-26178	Fix version increment when importing text file to create 'Note' document	API change	<p>All <code>FileManager#createDocumentFromBlob</code> methods have been deprecated, you should now use <code>FileManager#createOrUpdateDocument(FileImporterContext)</code> method to create or update documents from blobs.</p> <p><code>FileImporter#create</code> method has been deprecated, you should use <code>FileImporter#createOrUpdate(FileImporterContext)</code> method.</p> <p>The <code>FileImporterContext</code> class contains everything needed to create a document from a blob, see https://github.com/nuxeo/nuxeo/blob/release-10.10/nuxeo-services/nuxeo-platform-filemanager-api/src/main/java/org/nuxeo/ecm/platform/filemanager/api/FileImporterContext.java.</p> <p>Before:</p> <pre>DocumentModel doc = fileManager.createDocumentFromBlob(session, blob, parentPath, /* overwrite */ true, /* filename */ "foo.html");</pre> <p>After:</p> <pre>FileImporterContext context = FileImporterContext.builder(session, blob, parentPath) .overwrite(true) .fileName("foo.html") .build(); DocumentModel doc = fileManager.createOrUpdateDocument(context);</pre>

Key	Summary	Impact type	Upgrade notes
			<p>The <code>FileImporterContext#getBlob</code>'s file name is used by default now, but you can still override it with <code>FileImporterContext.Builder#fileName(String)</code>.</p> <p>If the underlying <code>FileImporter</code> handles it, there is now a way to let the <code>FileManager</code> create a document from a blob without persisting it using <code>FileImporterContext.Builder#persistDocument(boolean)</code>. It's then the caller's responsibility to actually persist the document.</p> <pre>FileImporterContext fileCreationContext = FileImporterContext.builder(session, blob, parentPath) .persistDocument(false) .build(); DocumentModel doc = fileManager.createOrUpdateDocument(fileCreationContext); // doc is not persisted // doc.isDirty() => true // doc.getId() => null // do something on doc doc = session.createDocument(doc); // doc is persisted</pre> <p>A new atomic operation <code>FileManager.ImportWithProperties</code> has been added allowing to create a document from a blob and setting properties at the same time. See https://github.com/nuxeo/nuxeo/blob/release-10.10/nuxeo-features/nuxeo-automation/nuxeo-automation-features/src/main/java/org/nuxeo/ecm/automation/core/operations/services/FileManagerImportWithProperties.java</p> <h3>Breaking Change</h3> <p>Custom importers implementing <code>FileImporter</code> or extending <code>AbstractFileImport</code> must be updated to override the <code>#createOrUpdate(FileImporterContext)</code> method instead of the <code>#create</code> method. Not overriding the <code>#createOrUpdate(FileImporterContext)</code> method will make your custom importer useless as it will never be called.</p> <p>See this commit for a sample of migrating to <code>#createOrUpdate(FileImporterContext)</code>.</p>
NXP-25617	Update mongodb template to handle KeyValueStore and LockManager		<p>Since 10.3 the mongodb template set the implementation of the <code>keyValueStore</code> and <code>LockManager</code> to MongoDB.</p> <p>If you want to keep or use a Redis implementation you need to be explicit in your <code>nuxeo.conf</code>:</p> <pre>nuxeo.keyvalue.provider=redis nuxeo.lock.manager=redis</pre>

Key	Summary	Impact type	Upgrade notes
NXP-25928	Repository LIKE queries should be anchored on MongoDB	Configuration Change	<p>For MongoDB queries, the LIKE queries are now anchored by default on master. This can be reverted to the old behavior using:</p> <pre><require>org.nuxeo.ecm.core.storage.mongodb.configuration</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <documentation> Property to disable the fact that on MongoDB a LIKE match anchors its search (like in SQL databases), i.e., absent wildcards, it does not match in the middle of a string. When true, does not match in the middle of a string. When false, works as if there was a % on each side of the LIKE expression. @since 10.3 </documentation> <!-- default is false for 9.10 and earlier --> <property name="nuxeo.mongodb.like.anchored">false</property> </extension></pre> <p>Conversely, this can be set to <code>true</code> on 9.10 if anchoring is required.</p>
NXP-26072	Enable use of a custom MongoDB Client TrustStore for in-flight encryption	Configuration Change	<p>The following <code>nuxeo.conf</code> properties can be set to define appropriate TLS/SSL configuration for MongoDB:</p> <ul style="list-style-type: none"> ▪ <code>nuxeo.mongodb.ssl=true</code> ▪ <code>nuxeo.mongodb.truststore.path</code> ▪ <code>nuxeo.mongodb.truststore.password</code> ▪ <code>nuxeo.mongodb.truststore.type</code> ▪ <code>nuxeo.mongodb.keystore.path</code> ▪ <code>nuxeo.mongodb.keystore.password</code> ▪ <code>nuxeo.mongodb.keystore.type</code> <p>If more fine-grained configuration is needed than properties, the following extension point can be used instead:</p> <pre><require>org.nuxeo.mongodb.connection</require> <extension target="org.nuxeo.runtime.mongodb.MongoDBComponent" point="connection"> <connection id="default"> ... <ssl>true</ssl> <trustStorePath>/path/to/cacerts.jks</trustStorePath> <trustStorePassword>changeit</trustStorePassword> <trustStoreType>jks</trustStoreType> <keyStorePath>/path/to/keystore.jks</keyStorePath> <keyStorePassword>changeit</keyStorePassword> <keyStoreType>jks</keyStoreType> </connection> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-17479	Create MySQL fulltext table with InnoDB		<p>Since Nuxeo 10.3, fulltext tables stored in MySQL/MariaDB now use InnoDB engine instead of MyISAM. InnoDB engine is more performant and allow to add foreign constraints on table's properties.</p> <p>You can change your current engine by following this documentation.</p> <p>Note: As there weren't any foreign constraints on these tables, you must first purge tables before converting them to new InnoDB engine with the constraint.</p> <p>You need to run:</p> <pre>DELETE FROM `fulltext` WHERE id not in (SELECT id FROM hierarchy);</pre>
NXP-25219	CSV Importer can't import Date properties with times		<p>Dates in CSV files must be formatted using the W3C format. More information on the W3C documentation.</p> <p>It is possible to use the old legacy date format (without time information) MM/dd/yyyy by setting the configuration property <code>nuxeo.csv.import.legacyDateFormat</code> to <code>true</code>:</p> <pre><require>org.nuxeo.ecm.csv.core.properties</require> <extension point="configuration" target="org.nuxeo.runtime.ConfigurationService"> <property name="nuxeo.csv.import.legacyDateFormat">true</property> </extension></pre>
NXP-26283	Backport NXP-25219 to 9.10		<p>Dates in CSV files are using a legacy date format (without time information) MM/dd/yyyy.</p> <p>However, it is possible to use the W3C date format, that supports time information, by setting the configuration property <code>nuxeo.csv.import.legacyDateFormat</code> to <code>false</code>:</p> <pre><require>org.nuxeo.ecm.csv.core.properties</require> <extension point="configuration" target="org.nuxeo.runtime.ConfigurationService"> <property name="nuxeo.csv.import.legacyDateFormat">false</property> </extension></pre> <p>More information on the W3C documentation.</p>
NXP-24316	Rework Video conversion listener for mass import		<p>Removed <code>VideoConstants#VIDEO_CHANGED_EVENT</code>, <code>VideoAutomaticConversionListener</code>, <code>VideoStoryboardListener</code></p> <p>Added <code>VideoInfoWork</code>, <code>VideoStoryboardWork</code></p> <p>The video info (duration, format, etc.) is now computed by an asynchronous work to avoid loading the blob and running <code>ffmpeg-info</code> synchronously. This work, in turn, schedules two asynchronous works to process the video storyboard and conversions.</p> <p>As a consequence, the user might not have the video info in the UI immediately after creating / updating a Video document, needing to refresh the page once the asynchronous work is done.</p>

Key	Summary	Impact type	Upgrade notes
NXP-25501	Fix nuxeo-dam Picture rendition as OriginalJpeg	API change	<p>Removed as deprecated since 7.2</p> <pre>RenditionServiceImpl#automationService RenditionServiceImpl#getAutomationService()</pre> <p>Deprecated</p> <pre>RenditionServiceImpl#getRenditionDefinition(String name) RenditionServiceImpl#storeRendition(DocumentModel sourceDocument, Rendition rendition)</pre> <p>Added</p> <pre>RenditionService#getAvailableRenditionDefinition(DocumentModel doc, String renditionName) RenditionServiceImpl#storeRendition(DocumentModel sourceDocument, Rendition rendition, RenditionDefinition renditionDefinition)</pre>

Key	Summary	Impact type	Upgrade notes
NXP-25208	hardcoded blacklisted mime types list prevents customizations in nuxeo-diff		<p>Deprecated <code>ContentDiffHelper#getHtmlConversionBlackListedMimeTypes</code>.</p> <p>Added:</p> <pre>org.nuxeo.ecm.diff.content.MimeTypesDescriptor org.nuxeo.ecm.diff.content.MimeTypeDescriptor ContentDiffAdapterManager#getHtmlConversionBlacklistedMimeTypes()</pre> <p>Added the <code>htmlConversionBlacklistedMimeTypes</code> extension point to the <code>org.nuxeo.ecm.diff.content.adapter.ContentDiffAdapterManagerComponent</code> component:</p> <pre><extension-point name="htmlConversionBlacklistedMimeTypes"> <documentation> @since 10.10 Allows to contribute the list of blacklisted mime types for HTML conversion. By default, contributing a list merges its mime types with the existing ones. To remove a mime type, use enabled=false. </code> <extension target="org.nuxeo.ecm.diff.content.adapter.ContentDiffAdapterManagerComponent" point="htmlConversionBlacklistedMimeTypes"> <mimeTypes> <contentType>application/msword</contentType> <contentType>application/rtf</contentType> <contentType enabled="false">application/pdf</contentType> </mimeTypes> </extension> </code></pre>
NXP-19262	Improved query API for directories	API change	<p>New directory query APIs using a <code>QueryBuilder</code>:</p> <ul style="list-style-type: none">▪ <code>Session.query(QueryBuilder, fetchReferences)</code>▪ <code>Session.queryIds(QueryBuilder)</code>

Key	Summary	Impact type	Upgrade notes
NXP-19264	Improved query API for UserManager	API change	<p>New UserManager query APIs using a QueryBuilder:</p> <ul style="list-style-type: none">▪ <code>UserManager.searchUsers(QueryBuilder)</code>▪ <code>UserManager.searchGroups(QueryBuilder)</code> <p>New MultiTenantUserManager query APIs like above but with a context:</p> <ul style="list-style-type: none">▪ <code>MultiTenantUserManager.searchUsers(QueryBuilder, context)</code>▪ <code>MultiTenantUserManager.searchGroups(QueryBuilder, context)</code> <p>New GroupComputer API using a QueryBuilder:</p> <ul style="list-style-type: none">▪ <code>GroupComputer.searchGroups(QueryBuilder)</code>

Key	Summary	Impact type	Upgrade notes
NXP-23863	Switch logger backend to Log4j 2		<p>Nuxeo now uses Log4j 2 as its logging backend instead of Log4j. Two logging APIs are generally available in Nuxeo:</p> <ul style="list-style-type: none">▪ Commons Logging▪ Log4j 2 API <p>SLF4J is still available, but Log4j 2 API are preferred by reason of its lambda support.</p> <p>Breaking change</p> <ul style="list-style-type: none">▪ <code>Log4jHelper.setQuiet</code> and <code>Log4jHelper.setDebug</code> was removed due to its usage of Log4j API which are now impossible with Log4j 2. These APIs were replaced by <code>Log4jHelper.setLevel</code>. Compared to previous implementation, <code>setLevel</code> API doesn't change appender's threshold, it just acts on logger's levels.▪ <code>LogCaptureFeature.Filter</code> has also changed without backward compatibility because it exposed a Log4j internal class. This has been replaced by corresponding Log4j 2 internal class.▪ File rollover has changed. Files are now compressed using gzip and pattern has been generalised to <code>NAME-YYYY-MM-dd.log.gz</code>. For instance: <pre>server.log.2018-01-01</pre> becomes <code>server-2018-01-01.log.gz</code>▪ <code>Watch log4j file for reload</code> has been removed in favor of built-in logic in Log4j2. These properties are no longer read:<ul style="list-style-type: none">▪ <code>org.nuxeo.runtime.log4jwatch.disabled</code>▪ <code>org.nuxeo.runtime.log4jwatch.delay</code>See <i>Automatic Reconfiguration</i> in Log4j 2 documentation to have watch file behavior. <p>Code behavior change</p> <ul style="list-style-type: none">▪ <code>LogCaptureFeature</code> behavior has changed. Now <code>log Result</code> is cleaned between each tests. You'll have an isolated result to assert for <code>@Before/@After</code>/test methods.▪ You may need to add the <code>slf4j</code> dependency if you were using it in your source code:<pre><dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> </dependency></pre> <p>New</p> <ul style="list-style-type: none">▪ <code>LogCaptureFeature.FilterOn</code> now accepts a logger class instead of a logger name▪ You can now use Log4j 2 API in your code

Key	Summary	Impact type	Upgrade notes
NXP-24030	Configure HSTS by default	Configuration Change	<p>When HTTPS is enabled (which is the case if a non-0 value is specified for <code>nuxeo.server.https.port</code>), HSTS is automatically enabled with the following defaults:</p> <ul style="list-style-type: none"> ▪ <code>nuxeo.server.hsts.maxage=2592000</code> ▪ <code>nuxeo.server.hsts.includesubdomains=false</code> ▪ <code>nuxeo.server.hsts.preload=false</code> <p>HSTS can be disabled by specifying:</p> <ul style="list-style-type: none"> ▪ <code>nuxeo.server.hsts.enabled=false</code>
NXP-26553	Use a template for Redis configuration	Configuration Change	<p>Activating Redis is now done through a template. Previously, it was enough to do:</p> <pre>nuxeo.redis.enabled=true</pre> <p>But now a <code>redis</code> template must be added instead:</p> <pre>nuxeo.templates=default,...,redis</pre>
NXP-24102	Upgrade Elasticsearch from 5.6.3 to 6.3.0		The Elasticsearch cluster must be upgraded to 6.x, customized elasticsearch settings and mapping may require update, follow the Nuxeo 10.2 upgrade procedure.
NXP-24114	<code>ecm:versionVersionableId</code> should be indexed by Elasticsearch		<p>With an Elasticsearch NXQL query you can retrieve all versions of a document by version series id. To search existing documents by <code>ecm:versionVersionableId</code> a re-index is required. This could either be via a full re-index or a re-index of just documents that have versions, using this query:</p> <pre>SELECT * FROM Document WHERE ecm:isVersion = 1</pre>
NXP-25214	Switch to Elasticsearch RestClient by default		<p>The <code>elasticsearch.addressList</code> option in <code>nuxeo.conf</code> needs to be updated to access elasticsearch in HTTP REST, the format is <code>http://elasticnode:9200</code> instead of <code>elasticnode:9300</code> used by the Transport client protocol. If you want to stay on the Transport client protocol, you have to explicitly set <code>elasticsearch.client=TransportClient</code> in your <code>nuxeo.conf</code>.</p>

Key	Summary	Impact type	Upgrade notes
NXP-26074	Enable use of a custom Elasticsearch RestClient Truststore	Configuration Change	<p>The following nuxeo.conf properties can be set to define appropriate TLS/SSL configuration for the Elasticsearch RestClient:</p> <ul style="list-style-type: none">▪ <code>elasticsearch.restClient.truststore.path</code>▪ <code>elasticsearch.restClient.truststore.password</code>▪ <code>elasticsearch.restClient.truststore.type</code>▪ <code>elasticsearch.restClient.keystore.path</code>▪ <code>elasticsearch.restClient.keystore.password</code>▪ <code>elasticsearch.restClient.keystore.type</code> <p>The following properties are deprecated (they were misnamed and are actually referring to the trustStore, not the keyStore):</p> <ul style="list-style-type: none">▪ <code>elasticsearch.restClient.keystorePath</code>▪ <code>elasticsearch.restClient.keystorePassword</code>▪ <code>elasticsearch.restClient.keystoreType</code> <p>If more fine-grained configuration is needed than properties, the following extension point can be used instead:</p> <pre><require>org.nuxeo.elasticsearch.defaultConfig</require> <extension target="org.nuxeo.elasticsearch.ElasticSearchComponent" point="elasticSearchClient"> <elasticSearchClient class="org.nuxeo.elasticsearch.client.ESRestClientFactory"> ... <option name="trustStorePath">/path/to/cacerts.jks</option> <option name="trustStorePassword">changeit</option> <option name="trustStoreType">jks</option> <option name="keyStorePath">/path/to/keystore.jks</option> <option name="keyStorePassword">changeit</option> <option name="keyStoreType">jks</option> </elasticSearchClient> </extension></pre>
NXP-25791	Make an Operation that launches Image Recompute		Added the <code>Picture.RecomputeViews</code> operation for administrators.

Key	Summary	Impact type	Upgrade notes
NXP-23744	UIDSequencer based on Key/Value Store	Configuration Change	<p>A new <code>KeyValueStoreUIDSequencer</code> is available, to store sequences in a key/value store. The store is the same for all sequencers, but they are using different keys, prefixed by the sequencer name.</p> <p>It can be used by doing for example:</p> <pre><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="configuration"> <!-- this is an example, "sequence" is already the default --> <property name="nuxeo.uidseq.keyvaluestore.name">sequence</property> </extension></pre> <p>Assuming the following configuration to define a suitable key/value store, for example:</p> <pre><extension target="org.nuxeo.runtime.kv.KeyValueService" point="configuration"> <store name="sequence" class="org.nuxeo.ecm.core.mongodb.kv.MongoDBKeyValueStore"> <property name="collection">sequence</property> </store> </extension></pre>
NXP-23745	Add increment API to Key/Value Store	API change	<p>New APIs are available on <code>KeyValueStore</code>.</p> <p>Optimized storage of Long values:</p> <ul style="list-style-type: none"> ▪ <code>put(String key, Long value)</code> ▪ <code>put(String key, Long value, long ttl)</code> ▪ <code>getLong(String key)</code> ▪ <code>getLongs(Collection<String> keys)</code> <p>Atomic increment:</p> <ul style="list-style-type: none"> ▪ <code>addAndGet(String key, long delta)</code>
NXP-24847	Allow configuring the blobProvider used by a KeyValueBlobTransientStore	Configuration Change	<p>A <code>KeyValueBlobTransientStore</code> can now specify the ids of the key/value store and blob provider to use, instead of defaulting to the name of the transient store itself:</p> <pre><extension target="org.nuxeo.ecm.core.transientstore.TransientStorageComponent" point="store"> <store name="mytransientstore" class="org.nuxeo.ecm.core.transientstore.keyvalueblob.KeyValueBlobTransientStore"> <property name="keyValueStore">mykeyvaluestore</property> <property name="blobProvider">myblobprovider</property> ... </store> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-26183	New KeyValueStoreProvider.keyStream(prefix) API	API change	New API: KeyValueStoreProvider.keyStream(prefix)
NXP-25837	Login screen config prevents SAML icon to be displayed	API change	<p>Added</p> <pre> LoginScreenHelper#registerSingleProviderLoginScreenConfig(String name, String iconUrl, String link, String label, String description, LoginProviderLinkComputer computer) LoginScreenHelper#unregisterLoginScreenConfig(LoginScreenConfig config) LoginProviderLink(String name, String iconPath, String link, String label, String description, LoginProviderLinkComputer urlComputer) LoginScreenConfig(LoginProviderLink provider) PluggableAuthenticationService#registerLoginScreenConfig(LoginScreenConfig config) PluggableAuthenticationService#unregisterLoginScreenConfig(LoginScreenConfig config) </pre> <p>Deprecated</p> <pre> LoginScreenHelper#registerLoginProvider(String name, String iconUrl, String link, String label, String description, LoginProviderLinkComputer computer) LoginScreenConfig#registerLoginProvider(String name, String iconUrl, String link, String label, String description, LoginProviderLinkComputer computer) </pre>
NXP-24885	Document Nuxeo Drive operations	API change	<p>Deprecated the following operations:</p> <pre> GetRepositoriesOperation NuxeoDriveAddToLocallyEditedCollection NuxeoDriveCanMove NuxeoDriveGenerateConflictedItemName NuxeoDriveGetClientUpdateInfo NuxeoDriveWaitForAsyncCompletion </pre> <p>Deprecated the following operation parameters:</p> <pre> NuxeoDriveGetChangeSummary#lastSyncActiveRootDefinitions </pre>

Key	Summary	Impact type	Upgrade notes
NXP-25797	Drive error when uploading zip file to server	API change	<p>Added to FileManager:</p> <pre>DocumentModel createDocumentFromBlob(CoreSession documentManager, Blob input, String path, boolean overwrite, String fullName, boolean noMimeTypeCheck, boolean excludeOneToMany) throws IOException;</pre> <p>to allow excluding "one-to-many" importers (ie. the ones creating more than one document for the given blob, typically CSVZipImporter or ExportedZipImporter) when selecting the importer. This is used by Drive to not try to explode a CSV import ZIP for instance.</p> <p>Added to FileImporter:</p> <pre>boolean isOneToMany();</pre> <p>to flag an importer as "one-to-many", ie. it creates more than one document for the given blob, typically CSVZipImporter or ExportedZipImporter.</p>

Key	Summary	Impact type	Upgrade notes
NXP-25844	Remove usage of deprecated code and code deprecated since 7.10 or less in nuxeo-drive-server	API change	<p>Removed as deprecated since 7.10 or less:</p> <pre> FileSystemItemFactory#getDocumentByFileSystemId(String id, Principal principal) FileSystemItemManager#getSession(String repositoryName, Principal principal) NuxeoDriveManager#setChangeFinder(FileSystemChangeFinder changeFinder) NuxeoDriveActions#UPDATE_SITE_URL_PROP_KEY NuxeoDriveActions#SERVER_VERSION_PROP_KEY NuxeoDriveCreateFile#name NuxeoDriveGetChangeSummary#lastSyncDate NuxeoDriveGetTopLevelChildren </pre> <p>Removed as should have been deprecated since 7.10 or less, see NXP-14826:</p> <pre> FileSystemChangeFinder#getFileSystemChangesIntegerBounds(CoreSession session, Set<IdRef> lastActiveRootRefs, SynchronizationRoots activeRoots, Set<String> collectionSyncRootMemberIds, long lowerBound, long upperBound, int limit) FileSystemChangeFinder#getCurrentDate(); NuxeoDriveManager#getChangeSummaryIntegerBounds(Principal principal, Map<String, Set<IdRef>> lastSyncRootRefs, long lowerBound) </pre> <p>and changed</p> <pre> FileSystemChangeFinder#getFileSystemChanges(CoreSession session, Set<IdRef> lastActiveRootRefs, SynchronizationRoots activeRoots, long lastSuccessfulSyncDate, long syncDate, int limit) NuxeoDriveManager#getChangeSummary(Principal principal, Map<String, Set<IdRef>> lastSyncRootRefs, long lastSuccessfulSync) </pre> <p>to</p> <pre> FileSystemChangeFinder#getFileSystemChanges(CoreSession session, Set<IdRef> lastActiveRootRefs, SynchronizationRoots activeRoots, Set<String> collectionSyncRootMemberIds, long lowerBound, long upperBound, int limit) NuxeoDriveManager#getChangeSummary(Principal principal, Map<String, Set<IdRef>> lastSyncRootRefs, long lowerBound) </pre>
NXP-24485	Add WEB_DETECTION to Nuxeo Vision (supported by Google)		<p>Added support for WEB_DETECTION tag when requesting Google Vision: https://cloud.google.com/vision/docs/reference/rest/v1/images/annotate#WebDetection</p>

Key	Summary	Impact type	Upgrade notes
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 predefined list. This allows for using any new feature available without waiting for an update of the plugin. After using such new feature, caller should use the <code>getNativeObject</code> method and handle the results based on the provider's documentation.
NXP-24121	nuxeoctl --strict does not work with startbg		Nuxeo Server will stop correctly in strict mode when startup failed.
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-26155	Improve query-related classes: Predicates/QueryBuilder/AST	API change	<code>Predicates</code> has many more useful static methods. <code>OrderByList</code> now has an empty constructor and a copy constructor. The <code>QueryBuilder</code> API has been clarified, and a <code>QueryBuilder</code> can internally hold a AND-based or OR-based <code>MultiExpression</code> . It can also hold a <code>countTotal</code> flag.
NXP-21017	Add an API to recompute quota for Tenant or User	API change	New operation: <code>Quotas.RecomputeStatistics</code> , with optional parameters: <ul style="list-style-type: none"> ▪ <code>tenantId / username / path</code> (only one allowed) ▪ <code>updaterName</code> (defaults to <code>documentsSizeUpdater</code>)
NXP-26073	Allow configuring Redis in SSL mode and with custom TrustStore	Configuration Change	The following <code>nuxeo.conf</code> properties can be set to define appropriate TLS/SSL configuration for Redis: <ul style="list-style-type: none"> ▪ <code>nuxeo.redis.ssl=true</code> ▪ <code>nuxeo.redis.truststore.path</code> ▪ <code>nuxeo.redis.truststore.password</code> ▪ <code>nuxeo.redis.truststore.type</code> ▪ <code>nuxeo.redis.keystore.path</code> ▪ <code>nuxeo.redis.keystore.password</code> ▪ <code>nuxeo.redis.keystore.type</code> <p>If more fine-grained configuration is needed than properties, the following extension point can be used instead:</p> <pre><require>org.nuxeo.ecm.core.redis.config</require> <extension target="org.nuxeo.ecm.core.redis" point="configuration"> <server> ... <ssl>true</ssl> <trustStorePath>/path/to/cacerts.jks</trustStorePath> <trustStorePassword>changeit</trustStorePassword> <trustStoreType>jks</trustStoreType> <keyStorePath>/path/to/keystore.jks</keyStorePath> <keyStorePassword>changeit</keyStorePassword> <keyStoreType>jks</keyStoreType> </server> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-23861	Raising an exception while using REST shows the whole stack trace		<p>Nuxeo version >= 10.1</p> <p>By default, the exception stack trace is not written in the returned response when using the REST API. If you want to have it back, you need to set the <code>nuxeo.conf</code> parameter <code>org.nuxeo.rest.stack.enable</code> to <code>true</code>. See https://doc.nuxeo.com/nxdoc/error-handling/ for more information.</p> <p>Nuxeo version < 10.1</p> <p>The exception stack trace is still written if the media type is <code>application/json+nxentity</code> but it can be disabled for security reason with the <code>nuxeo.rest.write.exception.stack.trace</code> configuration parameter, which is set to <code>true</code> by default.</p> <p>To disable it:</p> <pre><extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.rest.write.exception.stack.trace">false</property> </extension></pre>
NXP-25036	Make REST API endpoints consume and produce 'application/json' as content type	API change	<p>The REST API now produces and consumes only <code>application/json</code> as content type:</p> <ul style="list-style-type: none">▪ <code>application/json+nxentity</code> is now never returned as content type response; the server does not expect it as content type request.▪ <code>application/json+nxrequest</code> content type should not be used anymore. It still works but it's deprecated server side. <code>application/json</code> should be used instead when POSTing to automation.▪ <code>application/json+esentity</code> which was never used was also removed. <p>Breaking change:</p> <ul style="list-style-type: none">▪ the <code>org.nuxeo.ecm.automation.jaxrs.io.documents.JsonESDocumentWriter</code>, located in <code>nuxeo-automation-io</code> has moved to <code>org.nuxeo.elasticsearch.io.JsonESDocumentWriter</code>, located in <code>nuxeo-elasticsearch-core</code>. The <code>JsonESDocumentWriter</code> does not implement anymore <code>MessageBodyWriter<DocumentModel></code>, you may need to update your code.

Key	Summary	Impact type	Upgrade notes
NXP-26126	Make the "blob" enricher a BlobProperty enricher		<p>Enrichers can now apply to the blob type.</p> <p>For instance, to get the links to open a LiveConnect blob in all applications associated to its MIME type, you can use the <code>enrichers-blob: appLinks</code> enricher.</p> <p>Any blob property, e.g. <code>file:content</code>, will then be enriched in the following way:</p> <pre>{ "file:content": { "name": "...", "mime-type": "...", ..., "appLinks": [{ "appName": "...", "icon": "...", "link": "..." }, ...] } }</pre>
			<p>Note: since such a blob enricher applies to a <code>BlobProperty</code>, the backing class should extend <code>AbstractJsonEnricher<BlobProperty></code> as in the example below:</p> <pre>public class BlobAppLinksJsonEnricher extends AbstractJsonEnricher<BlobProperty> { @Override public void write(JsonGenerator jg, BlobProperty blobProperty) throws IOException { ... } }</pre>
NXP-22544	Rework <code>@LocalDeploy</code> and <code>NXRuntimeTestCase#deployTestContrib</code>		<p><code>@LocalDeploy</code> is now deprecated it must not be used anymore, use <code>@Deploy</code> instead.</p> <p>It was too permissive, and sometimes the replacement may force to adapt the content of the annotation.</p>

Key	Summary	Impact type	Upgrade notes
NXP-24366	Make @Deploy @Repeatable and handle it		<p>Usage of @Deploy has been made easier.</p> <p>Before</p> <pre>@Deploy({ "org.nuxeo.runtime.metrics", "org.nuxeo.ecm.platform.audit.api", "org.nuxeo.ecm.core.persistence", "org.nuxeo.ecm.platform.audit", "org.nuxeo.ecm.platform.uidgen.core", "org.nuxeo.elasticsearch.core" })</pre> <p>After</p> <pre>@Deploy("org.nuxeo.runtime.metrics") @Deploy("org.nuxeo.ecm.platform.audit.api") @Deploy("org.nuxeo.ecm.core.persistence") @Deploy("org.nuxeo.ecm.platform.audit") @Deploy("org.nuxeo.ecm.platform.uidgen.core") @Deploy("org.nuxeo.elasticsearch.core")</pre>
NXP-25402	Handle Duration deserialisation in descriptor		<p>You can now use Duration type into your descriptors. Nuxeo handles the JDK Duration format (PT10M, PT200S, PT0.500S) or a specific format for days, hours, minutes, seconds and milliseconds (10m, 200s, 500ms).</p>
NXP-25604	Provide a KeyValueStore implementation in SQL	API change	<p>A new Key/Value Store based on SQL is available. To configure a server to use it, use:</p> <pre><extension target="org.nuxeo.runtime.kv.KeyValueService" point="configuration"> <store name="default" class="org.nuxeo.ecm.core.storage.sql.kv.SQLKeyValueStore"> <property name="datasource">jdbc/nuxeo</property> <property name="table">kv</property> </store> </extension></pre> <p>Note that the PostgreSQL implementation requires the use of a version of PostgreSQL >= 9.6 to implement the put operation (PostgreSQL 11.x is the supported version with Nuxeo 10.10 anyway).</p>
NXP-24754	S3 Direct Upload - Switch Temp Credentials to STS:GetFederationToken	API change	
NXP-26576	Make S3 upload work with instance roles		<p>The S3 direct upload capability has been adapted to follow the AWS recommendations and to improve security using IAM Roles instead of User.</p> <p>The AWS and Nuxeo configuration required to unblock this feature is documented here: https://doc.nuxeo.com/nxdoc/amazon-s3-online-storage/#s3-direct-upload</p> <p>For backward-compatibility, in Nuxeo 9.10 it's possible to use an IAM User or an IAM Role.</p> <p>In Nuxeo 10.10 one has to use an IAM Role.</p>
NXP-24766	Introduce clock skew in SAML authenticator		<p>This introduces a new <code>nuxeo.sam12.skewTimeMs</code> configuration property to control the clock skew in milliseconds. Default value is 60 * 1000 (1 minute).</p>

Key	Summary	Impact type	Upgrade notes
NXP-16398	Review jsessionid management in URL system		<p>Since 10.2, Tomcat is configured by default with the <code>COOKIE</code> session tracking mode. This prevents Tomcat from appending <code>jsessionid</code> to the URLs, for example a file download URL.</p> <p>In 10.2 and later, to disable the <code>COOKIE</code> tracking mode and keep Tomcat's out of the box configuration, set the following property in <code>nuxeo.conf</code>:</p> <pre>session.config.tracking.mode.cookie=false</pre> <p>In 9.10-HF12, 8.10-HF33 and 7.10-HF43, the previous behavior is kept. The new behavior can be enabled by setting</p> <pre>session.config.tracking.mode.cookie=true</pre> <p>Session tracking mode implications</p> <p>If the <code>COOKIE</code> mode is:</p> <ul style="list-style-type: none">▪ enabled: the <code>jsessionid</code> parameter will never be appended to the URLs. Yet, cookies need to be enabled in the browser.▪ disabled: the <code>jsessionid</code> parameter might be appended to some URLs, for instance when 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.
NXP-24682	duplicate values shown when only one field should		<p>The <code>item</code> parameter previously used by <code>user_group_display_template.xhtml</code> in an implicit way (not documented) is replaced by the <code>data</code> parameter to avoid any confusion with an existing variable named <code>item</code>, typically an iteration variable of a <code><ui:repeat></code> tag.</p> <p>As stated by the widget template's documentation, the <code>data</code> parameter is "the field value to display in case <code>info.type</code> is empty".</p> <p>It can be used this way:</p> <pre><ui:include src="user_group_display_template.xhtml"> <ui:param name="data" value="#{valueToDisplay}" /> </ui:include></pre> <p>See for instance:</p> <ul style="list-style-type: none">▪ single_user_suggestion_widget_template.xhtml▪ user_suggestion_widget_template.xhtml
NXP-25780	Revise the implementation of EditableModelImpl.getRowData		<p>Since 10.3 and 9.10-HF19, a new configuration property is exposed to allow an alternative management of a missing row. Use the following contribution to enable it:</p> <pre><require>org.nuxeo.ecm.platform.ui.web.configuration.default</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.jsf.skipMissingRow">true</property> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-26043	Reset selection when navigating away	Configuration Change	<p>From 10.3, the list CURRENT_SELECTION is reset when switching between main tabs. In 9.10, the following contribution is needed to reset the selection:</p> <pre><require>org.nuxeo.ecm.webapp.documentsLists.DefaultDocumentsLists</require> <extension target="org.nuxeo.ecm.webapp.documentsLists.DocumentsListsService" point="list"> <documentsList name="CURRENT_SELECTION"> <events> <event>folderishDocumentSelectionChanged</event> <event>searchPerformed</event> <event>mainTabsChanged</event> </events> <isSession>false</isSession> </documentsList> </extension></pre>
NXP-25680	Use a condition to protect Nuxeo from a CSRF attack from local filesystem	Configuration Change	<p>On 10.3, the behavior is still disallowed but will be allowed in dev mode. On 7.10, 8.10 and 9.10, the behavior is now again allowed by default. To prevent local filesystem files from being allowed to POST, use the following contribution:</p> <pre><require>org.nuxeo.ecm.platform.ui.web.cors</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.cors.allowNullOrigin">false</property> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-25828	Improve external permissions experience dealing with multiple items		<p>Since 8.10-HF40 and 9.10-HF24</p> <p>A new configuration property <code>nuxeo.transient.username.unique</code> is available to define if a computed transient username should be unique no matter what base username is provided, or if it should be always the same for a given base username.</p> <p>Having always the same transient username for a given base username allows to generate only one token for a given email when giving permission to an external user: that means if you invite the same external user on 2 documents, he won't have to log out from the first document to see the second one.</p> <p>It defaults to <code>true</code> for backward compatibility.</p> <p>To disable the uniqueness of the transient username computation, use the following contribution:</p> <pre><require>org.nuxeo.ecm.core.api.properties</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.transient.username.unique">false</property> </extension></pre> <p>Since 10.3</p> <p>A computed transient username is now always the same for a given base username, so only one token is generated for a given email when giving permission to an external user: that means if you invite the same external user on 2 documents, he won't have to log out from the first document to see the second one.</p> <p>To enable back the uniqueness of the transient username computation, use the following contribution:</p> <pre><require>org.nuxeo.ecm.core.api.properties</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.transient.username.unique">true</property> </extension></pre>
NXP-25087	Upgrade Kafka from 1.0.0 to 1.1.0		In addition to Kafka 1.0.0 cluster we also support Kafka 1.1.0 cluster.
NXP-24945	Sanitize tag name at creation time		<p>In addition to the characters ' (single quote), \ (backslash), % (percent) and space, that were already forbidden in a tag, the character / (forward slash) is now also forbidden.</p> <p>All these characters are removed at creation time and when we try to get a tag or update it.</p>
NXP-25035	Add configuration property to be able to reduce tag sanitisation on faceted implementation		A new configuration property <code>nuxeo.tag.sanitization.enabled</code> has been added to enable/disable tag sanitization (true by default)
NXP-25481	Make the UI reflect the changes when a tag is sanitized		New global function <code>String.sanitize(boolean)</code> to cleanup tags

Key	Summary	Impact type	Upgrade notes
NXP-24493	Use Tomcat instead of old Jetty 6 for tests		<p>Various Jetty* test classes are renamed to ServletContainer*:</p> <ul style="list-style-type: none">▪ JettyFeature -> ServletContainerFeature▪ JettyTransactionalFeature -> ServletContainerTransactionalFeature▪ @Jetty(port = ...) -> @ServletContainer(port = ...) <p>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 context attribute:</p> <pre><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></pre> <pre><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></pre> <pre><extension target="org.nuxeo.runtime.server" point="listener"> <listener context="/"> <listener-class>org.nuxeo.example.MyListener</listener-class> </listener> </extension></pre>

Key	Summary	Impact type	Upgrade notes
NXP-24717	Thumbnail should have a better resolution for non image files	API change	<p>In ThumbnailDocumentConverter, removed unnecessary constants:</p> <pre> THUMBNAIIL_CONVERTER_NAME THUMBNAIIL_SIZE_PARAMETER_NAME // use ThumbnailConstants#THUMBNAIIL_SIZE_PARAMETER_NAME instead THUMBNAIIL_DEFAULT_SIZE // use ThumbnailConstants#THUMBNAIIL_DEFAULT_SIZE instead </pre>
NXP-26282	Create feature "Regenerate thumbnails on demand"		<p>Added the RecomputeThumbnails operation for administrators.</p> <p>Sample call:</p> <pre> curl -v -H 'Content-Type:application/json' -d '{"params": {"query": "SELECT * FROM Document WHERE ecm:mixinType = \"Thumbnail\" AND thumb:thumbnail/data IS NULL AND ecm:isVersion = 0 AND ecm:isProxy = 0 AND ecm:isTrashed = 0"}, "context": {}}' -u Administrator:Administrator http://localhost:8080/nuxeo/site/automation/RecomputeThumbnails </pre>
NXP-24331	CSRF Protection for Platform	Configuration Change	<p>CSRF protection is activated by default and based on the CORS configuration and its allowOrigin and supportedMethods parameters, which by default doesn't allow any cross origin.</p> <p>To activate an insecure configuration that allows any cross origin, use:</p> <pre> <extension target="org.nuxeo.ecm.platform.web.common.requestcontroller.service.RequestControllerService" point="corsConfig"> <!-- THIS IS INSECURE --> <corsConfig name="insecure" allowOrigin="*" supportedMethods="GET,HEAD,OPTIONS,POST,PUT,DELETE" > <pattern>/.*</pattern> </corsConfig> </extension> </pre> <p>See https://doc.nuxeo.com/nxdoc/cross-origin-resource-sharing-cors/ for more.</p>
NXP-25966	Upgrade Restlet libraries to latest 2.4.1	API change	<p>Some Restlet-related imports have slightly changed packages. The Restlet maven artifacts also have been renamed slightly.</p>
NXP-25903	CSRF Token for Platform	Configuration Change	<p>Activation</p> <p>To activate CSRF Token verification, use the following configuration:</p> <pre> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.csrf.token.enabled">true</property> </extension> </pre> <p>When this is activated, all clients accessing Nuxeo will need to get a token and provide it on all requests that are not GET/HEAD.</p> <p>Note that even when the CSRF Token is not activated, other CSRF checks not using a token are still being done (using the Origin/Referer headers).</p>

Key	Summary	Impact type	Upgrade notes
			<p data-bbox="940 71 1243 103">Getting the token initially</p> <p data-bbox="940 119 1803 151">The client must use the following request with the header <code>CSRF-Token: fetch</code>:</p> <pre data-bbox="940 167 2150 247">GET /nuxeo CSRF-Token: fetch</pre> <p data-bbox="940 263 1467 295">The response will contain the token in the header:</p> <pre data-bbox="940 311 2150 391">200 OK CSRF-Token: uNTIwv3oEIb3singqJKSuJDNjM9ldVOjnwtxmFh</pre> <p data-bbox="940 438 1153 470">Passing the token</p> <p data-bbox="940 486 2139 550">Then on every request that is not a GET/HEAD (so this applies to POST/PUT/DELETE/etc.) the client must provide the same token, either in the <code>CSRF-Token</code> request header or in the <code>csrf-token</code> request parameter:</p> <pre data-bbox="940 566 2150 646">POST /nuxeo/something CSRF-Token: uNTIwv3oEIb3singqJKSuJDNjM9ldVOjnwtxmFh</pre> <p data-bbox="940 662 974 694">or</p> <pre data-bbox="940 710 2150 774">POST /nuxeo/something?csrf-token=uNTIwv3oEIb3singqJKSuJDNjM9ldVOjnwtxmFh</pre> <p data-bbox="940 813 1332 845">Missing, expired or invalid token</p> <p data-bbox="940 861 2049 925">If the token is missing, expired or invalid, the client will get a <code>403 Forbidden</code> error, and a <code>CSRF-Token: invalid</code> header will be set:</p> <pre data-bbox="940 941 2150 1021">403 Forbidden CSRF-Token: invalid</pre> <p data-bbox="940 1069 1265 1101">Skipping certain endpoints</p> <p data-bbox="940 1117 2116 1181">Some authentication endpoints need to be available with a <code>POST</code> without CSRF token checks. This can be done using for example:</p> <pre data-bbox="940 1197 2150 1308"><extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="nuxeo.csrf.token.skip" list="true"/>/login</property> </extension></pre>
NXP-26060	Fix merge of DocumentsList descriptor	Configuration Change	<p data-bbox="940 1340 1915 1372">Since 10.3, the events in a new contribution overrides the one from the default contribution.</p> <p data-bbox="940 1388 1254 1420">Previously they were ignored.</p>

Key	Summary	Impact type	Upgrade notes
NXP-24434	Configure document type main publish rendition	Configuration Change	<p>Default rendition computation script must now be defined per reason.</p> <p>For instance:</p> <pre><defaultRendition reason="download"> <script language="JavaScript"> function run() { return 'mainBlob' } </script> </defaultRendition></pre> <p>or</p> <pre><defaultRendition reason="publish"> <script language="JavaScript"> function run() { return 'pdf' } </script> </defaultRendition></pre>
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 order to make it work
NXP-24683	Fix tasks list overflow		It was fixed for 10.2
NXP-24880	Make it possible to disable aggregation computations on initial search		<p>A nuxeo search form result layout can dynamically tell whether aggregations should be computed. For instance, on the nuxeo-default-search-form.html, if aggregations should not be computed if the full text parameter is not set, you can add the following</p> <pre>skipAggregates: { type: Boolean, notify: true, computed: '_skipAggregates(params.*)' }</pre> <p>where</p> <pre>_skipAggregates : function() { return !this.params !this.params.ecm_fulltext this.params.ecm_fulltext.length === 0; },</pre>
NXP-25008	Backport Trash Flow		Trash management is now available in Web UI on 9.10 (from HF09) but is disabled by default. See the needed slot contributions in the comment to enable it.
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 endpoint to skip any aggregation computation on a page provider query.

Key	Summary	Impact type	Upgrade notes
NXP-25160	Fix bulk nuxeo-delete-documents-action availability		The bulk permanent delete action when browsing the trash pill is not available anymore. You can either use the empty trash action or navigate to the trashed document to permanently delete it from the document info bar.
NXP-25204	Nuxeo instance useless after default-domain is removed		User's workspaces stored under a trashed domain are no longer used. Trashing the default domain will kind of trashed the User's workspace. But they can also be restored if the default domain is restored.
NXP-25323	Move page.js back to routing.html	Configuration Change	If index.jsp page is overridden the page.js import should be removed.
NXP-25385	Review all Web UI dynamic and static resources http caching strategy		<p>Note that <code>filterConfig <pattern></code> contributions now match the full path and query parameter of the URI (previously the query parameter was not included).</p> <p>Thumbnails, blobs and nxfile resources can have the doc changeToken appended as a query params for optimized http caching.</p> <pre> <filterConfig name="cachedBPR" cached="true" private="true" cacheTime="31536000" transactional="true"> <!-- if url contains doc changeToken: approximately one year --> <pattern>\${org.nuxeo.ecm.contextPath}/.*@(blob preview rendition).*\?.*changeToken=.+</pattern> </filterConfig> <filterConfig name="cachednxfile" transactional="false" synchronize="false" cached="true" private="true" cacheTime="31536000"> <!-- if url contains doc changeToken: approximately one year --> <!-- transaction is manually handled in DownloadServlet --> <pattern>\${org.nuxeo.ecm.contextPath}/nxfile/*\?.*changeToken=.+</pattern> </filterConfig> </pre> <p>Otherwise, we let eTag play.</p>
NXP-25464	Upgrade paper-input to include fixes		<p>Unlocking paper-input version. At the time ticket is done, we pull `paper-input` `2.3.3` of which the inner native input does not have the 'nativeInput' html tag id anymore. See diff</p> <p>This will impact elements using `paper-input` and manipulating the inner native input with a `#nativeInput` css selector.</p>

Key	Summary	Impact type	Upgrade notes
NXP-25512	WebUI: Allow to override/extend fetch properties and enrichers used to browse a document		<p>You can now add new enrichers used when serializing the current document by contributing:</p> <pre><require>org.nuxeo.web.ui.properties.contrib</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="org.nuxeo.web.ui.enrichers">myEnricher</property> </extension></pre> <p>You can now override the fetched properties when loading the current document (default behavior is still resolving all properties):</p> <pre><require>org.nuxeo.web.ui.properties.contrib</require> <extension target="org.nuxeo.runtime.ConfigurationService" point="configuration"> <property name="org.nuxeo.web.ui.fetch.document" list="true" override="true">dc:creator</property> <property name="org.nuxeo.web.ui.fetch.document" >dc:nature</property> <property name="org.nuxeo.web.ui.fetch.document" >dc:coverage</property> <property name="org.nuxeo.web.ui.fetch.document" >dc:subjects</property> </extension></pre> <p>(see also NXP-26038)</p>
NXP-26184	Extract nuxeo-document-content logic to a behavior	API change	
NXP-26451	Allow the usage of nuxeo-dropzone for a blob list without "auto submit"		<ul style="list-style-type: none"> ▪ update-document flag is now what defines if the dropzone will auto submit /save the files added; ▪ with this change it is now possible to use dropzone for blob lists in a an edit layout. For this kind of scenarios is recommended to use also batch-append flag;
ELEMENTS-484	Add a default time option to nuxeo-date-picker		<p>You can now specify a default time for the picked-up date. Format is HH:mm:ss e.g. 12:45:23. Default is 00:00:00 (midnight).</p>