

Module Archivage

Le portage initial en v.6 de ce module a été réalisé à partir de la v.5.2.38-1 (iso-fonctionnel), avec le strict minimum d'adaptations nécessaires pour son intégration.

Toutefois, certaines fonctionnalités ont été repensées pour adresser la problématique de migration et de continuité des bases d'archives créées en v.5.

Il n'est disponible en v.6 qu'à partir de la **6.11.3**

Principe de fonctionnement

L'objet de ce module est d'extraire de la base socle de Workey les documents qui sont arrivés en fin de vie au regard de leur Workflow respectifs.

Les deux composantes principales du module sont:

1. **L'agent d'archivage**

Il s'agit d'un Agent Workey spécifique, dont l'action est de créer les bases d'archives, d'identifier les processus à archiver et de réaliser l'extraction des documents éligibles à l'archivage.

Pour chaque processus archivé, il insère dans la base d'archive désignée les documents extraits et les supprime de la base socle de Workey.

2. **L'application de consultation des archives**

Il s'agit de l'interface web permettant de rechercher et de visualiser les documents contenus dans les différentes bases d'archives.

Limitations

Attention

Il n'est pas possible de fusionner deux bases archives.

L'archivage d'un processus donné (et de toutes ses versions) ne peut se faire que vers une seule base d'archive désignée.

À éviter

Ne pas désigner une même base pour des processus différents !

Pré-requis

Le module étant désormais intégré au livrable global du moteur, son accès et sa mise en œuvre sont subordonnés à l'activation via le fichier de licence du serveur Workey.

Information

En cas de mise à niveau du serveur Workey depuis une version antérieure à la v.6, il sera indispensable de mettre à jour le fichier de licence, afin qu'y soit activée la fonctionnalité d'archivage. A l'occasion de ce portage, le générateur de licence a lui aussi été mis à jour.

Modélisation

Formulaire d'archivage

Depuis la version **5.2.33** de Workey, l'archivage requiert obligatoirement que chaque type document d'un processus dispose d'un Formulaire d'archivage.

Information

La définition de ce Formulaire est primordiale puisque celui-ci sera utilisé pour l'accès au document en vue de son archivage. Seules les données exposées par ce Formulaire seront exportées vers l'archive.

Attention

Corollaire : toute les autres données du document seront définitivement perdues.

Lors de la modélisation d'un Formulaire, il est possible de spécifier si celui-ci sera utilisé comme Formulaire d'écriture et/ou de lecture par défaut, mais aussi comme Formulaire d'archivage.

Nom	A	L	E
Formulaire ARCHIVE	☒	☐	☐
Formulaire RO	☐	☒	☐
Formulaire RW	☐	☐	☒

Il donc possible de définir un Formulaire exclusivement réservé à l'archivage

Formulaires

Nom	A	L	E
Mon Formulaire	☒	☒	☒

ou bien un unique Formulaire de Lecture – Écriture – Archivage

Information

Il ne peut y avoir qu'un seul Formulaire d'archivage pour un type de document donné.

Pour les processus déployés dans des versions antérieures à Workey v.6, il sera nécessaire de définir ces formulaires à posteriori, **directement en base**. Pour se faire il faudra identifier, pour chaque type de document et pour chacune des versions du processus, un Formulaire existant (dans la table FORMS) et de valoriser sa colonne IS_FOR_ARCHIVE à 1 ou TRUE (selon le SGBD).

Exemple : soit une base socle ne contenant absolument aucun formulaire, alors désigner les formulaires par défaut (à la fois en Lecture et en Écriture) comme étant les formulaires d'archivage.

```
-- Exemple pour MySQL
UPDATE FORMS SET IS_FOR_ARCHIVE=1 WHERE IS_READ_DEFAULT=1 AND
IS_WRITE_DEFAULT=1;
```

Limitations de la modélisation du formulaire d'archivage

En 6.11.3, certains composants de formulaire ne sont pas supportés lors de la visualisation des données archivées:

- Script JS
- Dossier MG et Documents MG

Certaines caractéristiques ne seront pas prises en compte:

- L'option "Vignette" des champs pièce-jointe
- Le nombre de colonne d'une table

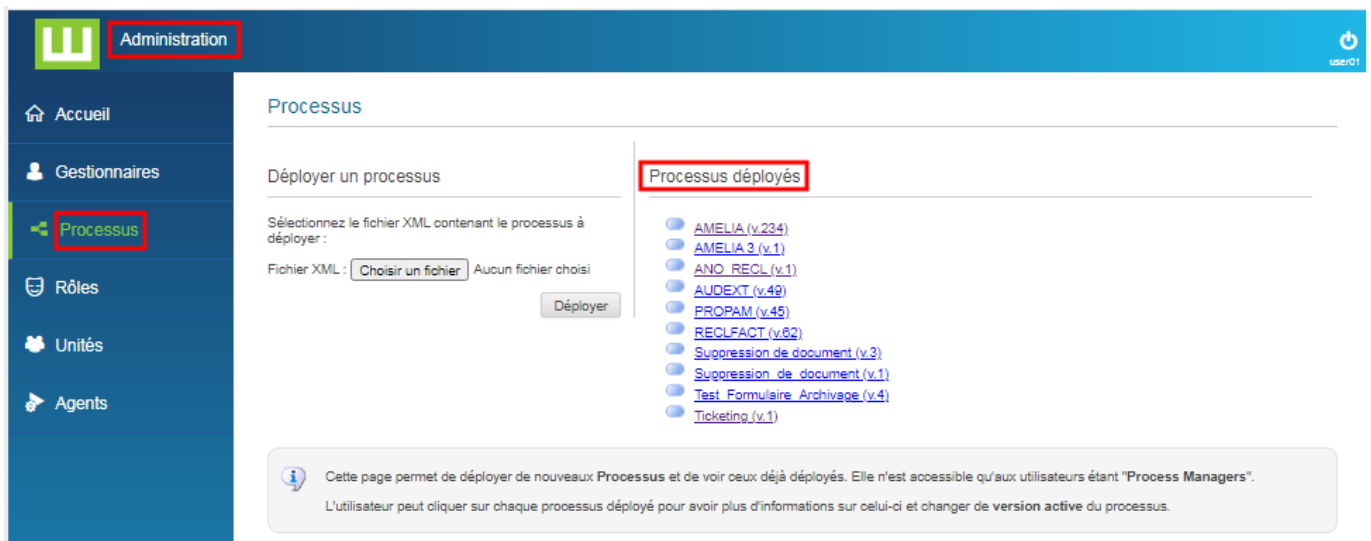
Mise en œuvre

La mise en œuvre du module d'archivage consiste en :

- la désignation des processus à archiver
- la définition d'un agent Workey spécifique

Processus à archiver

Depuis la console d'administration, dans la section "Processus", en cliquant sur un "Processus déployé", le détail de celui-ci s'affiche et notamment sa politique d'archivage.



Deux informations sont nécessaires pour que les documents d'un processus soient archivés :

- le nom de la base d'archive cible
- le délai d'archivage

Archivage

Nom de l'archive	ANOMALIES ET RECLAMATIONS
Délai d'archivage (en jours)	1000 (un délai négatif empêche les documents du processus d'être archivés)
Modifier l'archivage	

Note: toute modification apportée à ces paramètres doit être validée en cliquant sur le bouton «Modifier l'archivage» pour qu'elle soit effective.

Attention

Il peut arriver qu'une ou plusieurs versions inactives du processus soient susceptibles d'avoir des valeurs de paramétrage différentes de celles de la version active; l'interface le signalera et invitera l'utilisateur à revalider le paramétrage actuel – qui s'appliquera alors à l'ensemble des versions du processus – en cliquant sur « Modifier l'archivage ».

Archivage

Nom de l'archive
Délai d'archivage (en jours) (un délai négatif empêche les documents du processus d'être archivés)

 Les paramètres d'archivage ne sont pas identiques pour toutes les versions de ce processus. Vous devez re-définir ces valeurs.

[Modifier l'archivage](#)

Délai d'archivage

Un processus sera éligible à l'archivage si son délai d'archivage est **égal ou supérieur à 0**.

Ce délai est exprimé en **nombre de jour**.

Un **délai négatif** désactive l'archivage des documents de ce processus.

Lors du premier déploiement d'un processus, le délai d'archivage est par défaut fixé à 365 jours. Cette valeur peut toutefois être modifiée à posteriori par l'utilisateur. En cas de déploiement d'une nouvelle version du processus existant, alors le délai d'archivage déjà paramétré sera conservé.

Nom de la base d'archive

Les documents du processus archivé seront stockés dans la base d'archive désignée.

Lors du premier déploiement d'un processus, le nom de la base d'archive cible est déterminé depuis le nom interne du processus; les caractères d'espacement y sont substitués par des `_` (*underscore*). Ce nom peut toutefois être modifié par l'utilisateur. En cas de déploiement d'une nouvelle version du processus existant, alors le nom de la base d'archive déjà paramétré sera conservé.

Ce nom sera utilisé pour créer un sous-répertoire dans le répertoire global des archives.

Si la base d'archive telle que nommée, n'existe pas, sa création initiale sera assurée par l'agent d'archivage dès qu'un document du processus est archivé.

Attention:

- Le nom de la base d'archive ne doit pas contenir de caractère non autorisé pour la création d'un répertoire sur le système de fichier.
- Le nom ne peut être laissé vide ou constitué exclusivement de caractères d'espacement.
- Tout changement de nom de la base d'archive entraînera la création d'une nouvelle base; à moins de renommer manuellement le répertoire de la base existante, avant le prochain déclenchement de l'agent d'archivage.
- Le changement de base d'archive cible, entre deux déclenchements de l'agent d'archivage, impactera la continuité des archives car les documents nouvellement archivés seront alors stockés dans la nouvelle

base. Ceci peut s'avérer très problématique du fait qu'il **n'est pas possible de fusionner/consolider des bases d'archives** entre elles.

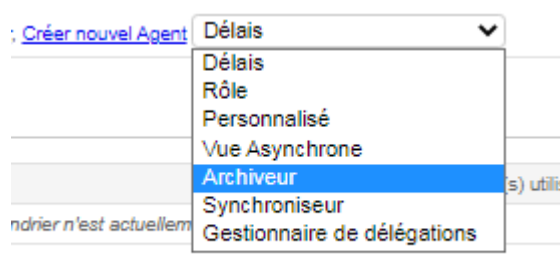
- Une base d'archive n'est pas *autonome*; en ce sens que la consultation des documents archivés qu'elle contient s'appuiera sur les droits d'accès propres aux acteurs authentifiés dans Workey.
- Pour la configuration du répertoire global des archives, se reporter à la documentation de la propriété de configuration du serveur `com.clog.workey.directory.archives`

Agent d'archivage

L'agent d'archivage est un type d'agent Workey spécifique. Lorsque la fonctionnalité d'archivage est activée au niveau de la licence, cet agent sera proposé dans la liste déroulante des types d'agent qui peuvent être créés.

Création de l'agent

Depuis la liste déroulante des types d'agent disponibles, sélectionner *Archiveur* et cliquer sur «Créer nouvel Agent».



De par son architecture, le module d'archivage repose sur l'action d'un **unique agent** spécifique. Il ne doit y avoir qu'un seul agent d'archivage qui scrute et traite l'ensemble des processus à archiver.

Information

Si un agent d'archivage a déjà été défini, mais que la fonctionnalité d'archivage est désactivée au niveau de la licence, alors l'agent se déclenchera au regard de sa programmation mais sera sans effet et s'interrompra aussitôt.

À éviter

Ne pas définir plusieurs agents d'archivage car il n'y aura pas de coordination entre les instances de ces différents agents et leurs actions respectives se contrarieraient lors du traitement des documents, aboutissant à une corruption des bases d'archive.

Ne pas définir un agent personnalisé en spécifiant la même classe java que celle de l'agent d'archivage.

Paramétrage de l'agent

L'écran de paramétrage initial de l'agent est le suivant.

Détail de l'Agent

Options génériques de l'Agent

- ☐ Journalisation de l'activité
☐ Autoriser les instances multiples

Tâche de l'Agent

Type de tâche: Archiveur
Description de la tâche: Archive tous les documents éligibles
Nom du répertoire d'archivage C:\Java\apache-tomcat-8.5.38\work\key\archives
Paramètre(s) de la tâche

Nom	Utilisé	Valeur	Description
archive.timeout	☐		'Timeout' pour la tâche d'archivage (exprimé en minutes; nombre entier > 0)
archive.lang	☐	aa ▼	La langue à utiliser pour l'archivage de document (exemple 'fr', 'en')

Identité de l'Agent

Utilisation d'un Acteur de type AGENT spécifique

Information de Déclenchement

Mode d'exécution ☒ Répétition à intervalle régulier ☐ Expression CRON
Intervalle de temps: heure ▼
Mode de démarrage ☒ Dès que possible ☐ Compte-à-rebours ☐ Heure fixe ☐ Date fixe

Calendrier d'Exclusion

☐ Utiliser calendrier ▼

Appliquer



Cette page détaille les paramètres d'un Agent.

Journalisation

L'activation de la **journalisation** de l'agent permettra d'exposer les traitements effectués lors de ses instanciations successives.

Date	Heure	Message
2021.03.15	15:11:30	Trigger[VOLATILE.ID#9@1615817488949]:End of process
2021.03.15	15:11:30	Skipping archiving of process type 'Ticketing' as it has a negative Archive Duration.
2021.03.15	15:11:30	Processed 0 document for process type 'Test_Formulaire_Archivage'
2021.03.15	15:11:30	Skipping archiving of process type 'Suppression_de_document' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'Suppression de document' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'RECLFACT' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'PROPAM' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'Process_bidon' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'AUDEXT' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'ANO_RECL' as it has a negative Archive Duration.
2021.03.15	15:11:29	Skipping archiving of process type 'AMELIA 3' as it has a negative Archive Duration.
2021.03.15	15:11:29	Processed 0 document for process type 'AMELIA'
2021.03.15	15:11:29	Trigger[VOLATILE.ID#9@1615817488949]:Agent starting.

Instanciation multiple

En cohérence avec le pré-requis d'unicité de l'agent, les **instances multiples** ne sont pas autorisées.

Paramètre(s) de la tâche

Le paramètre "Nom du répertoire d'archivage" indique, à titre informatif, les nom et chemin du répertoire d'archive global tels qu'ils sont configurés au niveau du serveur Workey. Ce paramètre ne peut pas être modifié au niveau de la configuration de l'agent.

Les paramètres suivants sont optionnels:

- `archive.timeout` permet de spécifier une durée maximale de traitement de l'agent à chacune de ses instanciations. Ce *timeout* est exprimé en minutes. Avant d'archiver un document éligible, l'agent s'assure que le temps imparti n'est pas dépassé. Si tel est le cas, alors il interrompra son traitement et abandonnera l'archivage du document et éventuellement de l'ensemble des documents de sa famille.
- `archive.lang` permet de spécifier une locale particulière lors de l'ouverture des documents à archiver. Bien que le module d'archivage supporte la localisation des processus, ce paramétrage peut avoir une incidence dans des cas très spécifiques de modélisation (ex: connecteur de champ retournant des valeurs déjà localisées et qui seront donc définitivement figées lors de l'archivage). En l'absence de définition de ce paramètre, la langue utilisée sera celle configurée par défaut au niveau du serveur Workey (cf. documentation de la propriété `com.clog.workey.engine.defaultLanguage` de configuration du serveur Workey).

Identité de l'Agent

L'agent d'archivage utilisera une identité *ad hoc* et spécifique, ayant des privilèges similaires au Gestionnaire de Workflow, pour traiter les documents éligibles. Il n'y a donc aucun paramétrage à effectuer.

Identité de l'Agent

Utilisation d'un Acteur de type AGENT spécifique
Acteur ID=1614, ayant pour clef `cn=Agent, task=ARCHIVE`

Une fois l'agent créé, la page de détail de l'agent mentionnera le détail de l'identité *ad hoc*.

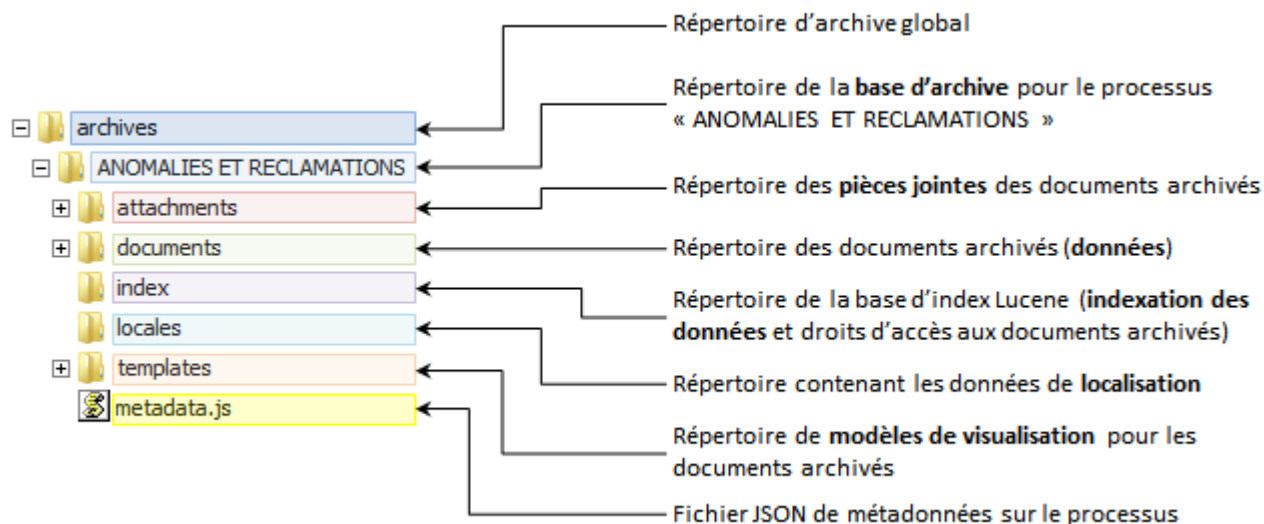
Information de Déclenchement et Calendrier d'Exclusion

Leurs paramétrages ne présentent aucune spécificité et s'inscrivent dans la même logique que pour les autres Agents Workey.

Structure d'une base d'archive

Les bases d'archives ont une structure commune.

Exemple: d'une base d'archive nommée "ANOMALIES ET RECLAMATION"



Répertoire à ventilation arborescente

Les répertoires attachment/ et documents/ présentent une structure particulière dont la gestion est assurée par le module d'archivage.



Le contenu effectif de ces répertoires sera toujours situé 2 sous-niveaux inférieurs. Ces niveaux intermédiaires sont constitués par des répertoires numérotés/nommés de 00 à FF (en hexadécimal). L'emplacement (chemin) d'un fichier donné sera déterminé au regard d'un algorithme de hachage à partir de son nom.

L'arborescence de répertoire sera créée automatiquement au fil de l'eau selon les besoins de l'agent d'archivage.

Cette stratégie de stockage a été motivée par la volumétrie potentiellement importante des archives.

Répertoire attachments/

C'est sous cette arborescence que seront stockées les pièces jointes des documents archivés.

Au moment de l'archivage, les fichiers correspondants aux pièces jointes sont recopiés vers de nouveaux fichiers avec la nomenclature spécifique suivante:

Concaténation:

- de l'UUID interne de la pièce jointe
- d'un caractère_ (*underscore*) de séparation
- du nom (avec son extension) du fichier d'origine **normalisé** de la pièce jointe

La normalisation du nom du fichier consiste en la «*désaccentuation* » (principe de ramener chaque caractère accentué à sa forme basique) et en la substitution des caractères d'espacement par des _ (*underscore*).

Exemple:

\attachments\b7\50\297e6ed476f1d4f00176f1f17dcb0000_Attestation_validee.pdf

Répertoire documents/

C'est sous cette arborescence que seront stockés les documents archivés avec leurs données.

Au moment de l'archivage, le document est sérialisé au format XML, via l'API Workey, vers un fichier avec la nomenclature suivante :

Concaténation:

- du terme document
- d'un caractère _ (*underscore*) de séparation
- de l'identifiant numérique interne du document
- de l'extension.xml

Exemple:

\documents\0d\0c\document_3773.xml

Répertoire index/

Ce répertoire contient les fichiers de la base *Lucene* utilisée pour indexer le contenu des documents présents dans l'archive, **ainsi que les droits d'accès à ces documents.**

L'indexation est assurée par la même brique fonctionnelle que celle du moteur Workey. Le détail des clefs indexées n'est pas couvert par la présente documentation.

Attention:

- Contrairement à la base *Lucene* adossée à la base socle de Workey, le

contenu de celle-ci ne peut être recréé à posteriori (pas de réindexation possible). En effet, elle est la seule à détenir les informations de droits d'accès aux document archivés !

- En cas de mise-à-niveau du serveur workey depuis une version antérieure à la v.6, il sera indispensable de migrer la base *Lucene* de chacune des bases d'archive précédemment créées. En effet, les bases d'archives créées sous Workey v.5 utilisaient *Lucene 2.9.4*. Le module d'archivage sous Workey v.6 utilise *Lucene 4.8.1*. La migration de structure de ces bases doit être effectuée en deux temps, à l'aide de la classes java `org.apache.lucene.index.IndexUpgrader`, prévue à cet effet par les concepteurs de *Lucene*:
 - migrer la structure des bases *Lucene* de 2.9.4 → 3.2.0 (nécessite de disposer de la librairie `lucene-core-3.2.0.jar` dans le classpath java)
 - puis migrer la structure des bases *Lucene* de 3.2.0 → 4.8.1 (nécessite de disposer de la librairie `lucene-core-4.8.1` dans le classpath java. Toutefois, celle-ci est déjà indirectement embarquée dans le fichier `workey.war` du livrable de Workey, sous `WEB-INF/lib`).

Exemple: des lignes de commande (DOS), pour migrer la base *Lucene* d'une base d'archive. Il est nécessaire de spécifier le chemin vers les fichiers de la base d'index

```
> java -cp lucene-core-3.2.0.jar org.apache.lucene.index.IndexUpgrader -  
verbose "ANOMALIES ET RECLAMATIONS\index"
```

```
> java -cp lucene-core-4.8.1.jar org.apache.lucene.index.IndexUpgrader -  
verbose "ANOMALIES ET RECLAMATIONS\index"
```

Répertoire locales/

Ce répertoire contient les **ressources de localisation** du processus archivé.

Ces ressources seront utilisées pour la visualisation des documents archivés lors de leur consultation par un utilisateur, au regard de sa préférence de langue.

Au moment de l'archivage, l'agent exportera les ressources de localisation pour chacune des langues supportées/configurées (cf. documentation de la propriété `com.clog.workey.engine.supportedLanguages` de configuration du serveur Workey), respectivement vers autant de fichiers selon la nomenclature suivante:

Concaténation:

- du terme `resources`
- d'un caractère `_` (*underscore*) de séparation
- de l'**identifiant numérique interne du type de processus**
- d'un caractère `_` (*underscore*) de séparation
- du code de la locale
- de l'extension `.xml`

Exemple:

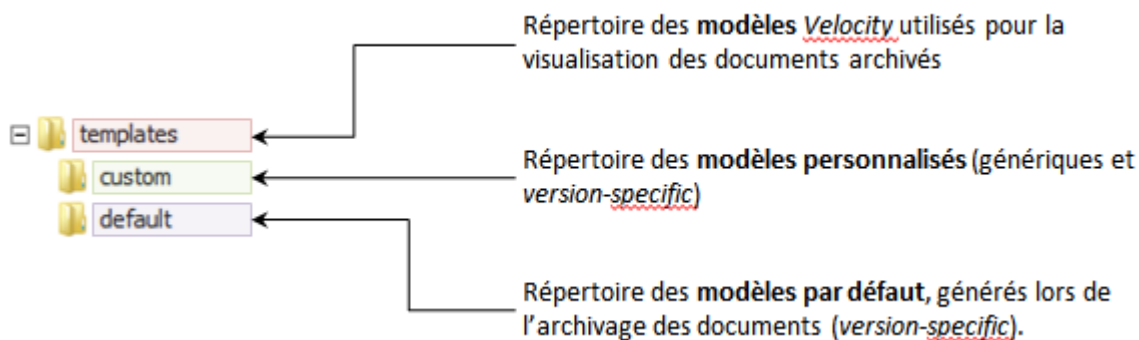
```
\locales\resources_16_en.xml  
\locales\resources_16_fr.xml  
\locales\resources_25_en.xml  
\locales\resources_25_fr.xml
```

Information

Si la liste de langues supportées venait à s'étoffer alors que des documents sont déjà archivés, alors la nouvelle localisation des ressources ne pourra être prise en compte qu'avec les documents d'une nouvelle version du processus.

Répertoire templates/

Ce répertoire contient les modèles (templates) *Velocity* utilisés pour restituer/visualiser les données des documents archivés. Ceux-ci sont désormais ventilés selon qu'ils sont personnalisés ou par défaut.



- custom/, répertoire contenant les templates personnalisés par l'utilisateur.
- default/, répertoire cible de la génération à la volée, par l'agent d'archivage, des templates (version-specific).

Chaque template est propre à un type de document. Toutefois, les templates peuvent être distingué en deux catégories: les **templates génériques** et les **templates spécifiques à une version**. Ils se différencient par la nomenclature des noms de fichier.

Un template générique a pour nom de fichier la concaténation:

- du nom interne du type de document
- de l'extension .vml

Un template spécifique à une version a pour nom la concaténation:

- du nom interne du type de document
- des caractères -v (trait d'union et la lettre v minuscule)
- du **numéro de version du type de processus**
- de l'extension .vml

Attention:

- En cas de mise-à-niveau du serveur worky depuis une version antérieure à la v.6:
 - l'agent d'archivage créera les deux répertoires custom/ et default/, s'ils n'existent pas déjà, lorsqu'un nouveau document sera archivé à destination de la base d'archive en question.
 - il sera nécessaire de déplacer les templates existant dans le répertoire correspondant en fonction des règles de précedence qui s'appliqueront (cf. [sélection des templates](#)).

Fichier metadata.js

```
{
  "process":"Test_Formulaire_Archivage",
  "delay":0,
  "documentTypes":[
    {
      "name":"Doc_Test_archive_form",
      "label":"Doc Test archive form"
    }
  ],
  "name":"Test_Formulaire_Archivage",
  "startTime":1614010213303,
  "endTime":1614010213603,
  "fields":[
    {
      "name":"Ligne_simple_3",
      "label":"Ligne simple 3"
    }, {
      "name":"Les_PJs",
      "label":"Les PJs"
    }, {
      "name":"Colonne_2",
      "label":"Colonne 2"
    }, {
      "name":"Colonne_1",
      "label":"Colonne 1"
    },
    {
      "name":"Derniere_colonne",
      "label":"Dernière colonne"
    }
  ]
}
```

Ce fichier contient la définition d'un objet JSON. Il est initialement généré, et complété au fil du traitement des documents, par l'agent d'archivage. Il contient les métadonnées relatives aux documents présents dans l'archive. Celles-ci seront exploitées par l'application de consultation des archives pour proposer un formulaire de recherche avancée.

- process, le nom interne du type de processus
- name, le nom de la base d'archive
- delay, le délai d'archivage appliqué à lors de la création initiale de l'archive
- documentTypes, un tableau JSON listant les types de document contenus dans l'archive, sous la forme de couples (nom interne, libellé):
 - name, le nom interne du type de document
 - label, le libellé du type de document
- startTime, horodatage de début de la dernière itération d'archivage
- endTime, horodatage de fin de la dernière itération d'archivage
- fields, un tableau JSON listant les champs des document présents dans l'archive, sous la forme de couples (nom interne, libellé):
 - name, le nom interne du champ
 - label, le libellé du champ

Archivage

Rappel: l'agent d'archivage scrute l'ensemble des processus de la base socle. Seuls les processus ayant un délai d'archivage supérieur ou égal à 0 jour sont pris en compte (PROCESS_TYPES.ARCHIVE_DURATION >= 0).

L'agent traite les processus éligibles dans l'ordre alphabétique de leur nom interne.

La structure de la base socle de Workey, avec ses contraintes relationnelles, impose que tous les documents liés par une relation père-fils (dérivation SANS démarrage d'un nouveau processus) doivent être archivés/extraits de la base simultanément (cf. Atomicité de l'archivage).

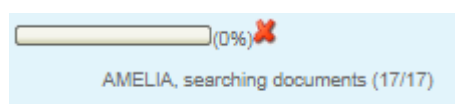
Pour chaque processus, l'agent commencera par dénombrer les familles de document (via leur document patriarche) qui répondent aux conditions d'éligibilité à l'archivage. Puis seulement après, il procèdera à l'archivage des documents **famille par famille**.

Les familles de document sont traitées dans l'ordre croissant des identifiants numériques de leur patriarche.

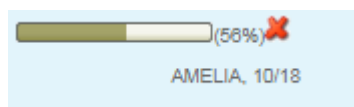
Information

La barre de progression de l'activité de l'agent d'archivage est réinitialisée entre chaque processus traité.

Le message de progression mentionne toujours le processus actuellement traité. Lors de la phase initiale de recherche des documents éligibles à l'archivage, le compteur indique le nombre de documents patriarches analysés; c'est-à-dire le nombre de famille distincte de document.



Dans un second temps, l'agent procède à l'archivage de l'ensemble des documents de ces familles. Le compteur indique alors le nombre de document actuellement traités. Bien entendu, ce nombre pourra être supérieur au nombre de patriarches dans la mesure où les familles peuvent être constituées de plus d'un document.



Éligibilité des documents

Tous les documents d'une même famille doivent répondre simultanément aux deux contraintes suivantes:

- le document doit **avoir atteint un état final**.
C'est-à-dire un état pour lequel il n'y a plus d'opération (modélisée) applicable. Le document est **clos**. Concrètement, dans la base socle, le document doit avoir son `WORKFLOW.NEXT_STEPS` ≤ 0 .
- le document doit avoir atteint son état final **depuis un laps de temps $\geq$ au délai d'archivage** défini pour le processus auquel il appartient. Pour cela, l'horodatage `WORKFLOW.LAST_ACCESS_TIME` du document sera confronté à la date courante.

Information

Il n'est pas possible d'exclure un type de document donné pour un processus archivé. Tous les types de documents seront pris en considération.

Atomicité de l'archivage

L'archivage d'une famille de document est un **traitement atomique**. Si l'archivage de l'un des documents échoue ou est interrompu, alors aucun des documents de la famille ne sera archivé. Le périmètre de ce traitement englobe les actions suivantes.

Vérification d'interruption

Avant d'entamer le traitement d'un document donné, il y a vérification des éventuels sémaphores d'interruption. **Toute interruption levée mettra fin à l'instance en cours de l'agent d'archivage.**

Limitation du nombre de document

A des fins de débogage, il est possible de définir un nombre maximum de document que l'agent d'archivage est autorisé à traiter à chaque déclenchement, via la propriété de configuration du serveur: `com.clog.workey.archive.debug.docLimit`.

Information

Cette limitation s'applique par processus.

Timeout global

Le paramétrage optionnel de l'agent permet de spécifier un temps de traitement global maximum pour une instanciation de l'agent (cf. paramètre `archive.timeout`).

Information

Cette limitation s'applique à l'instance de l'agent, quelque soit son avancée dans le traitement des différents processus.

Interruption à la demande de l'utilisateur

Depuis le tableau de bord des Agents dans la console d'administration du serveur Workey, il est possible d'interrompre l'instance en cours d'exécution de l'agent d'archivage en cliquant sur la croix rouge située à droite de sa barre de progression. Cette demande d'interruption est donc scrutée par l'agent avant de commencer à traiter un document.

Éviction des documents avortés

Si le document à traiter n'a jamais progressé dans le flux, alors il n'est pas archivé et sera supprimé. Typiquement, le document est resté dans l'état système `_CREATED` ou `_DERIVED`.

Parcours récursif arborescent

L'archivage d'une famille a pour point de départ le document patriarche. En revanche, l'archivage d'un document traite d'abord sa descendance avant d'archiver le document en lui même. Les documents fils sont traités dans l'ordre croissant de leur identifiant numérique. L'opération d'archivage prend donc la forme d'un traitement récursif du fait du parcours arborescent de la hiérarchie père-fils.

Accès au document pour archivage

Pour archiver un document, il est nécessaire de l'accéder en lecture à l'aide du Formulaire d'archivage défini pour ce type de document.

Le document est accédé avec les **privilèges d'un Gestionnaire de Workflow**.

À éviter

L'**absence** de formulaire d'archivage est **fatal** dans le traitement du document. L'instance de l'agent d'archivage sera alors **interrompue**.

Génération modèle *Velocity*

Information

Ce mécanisme est une refonte de la fonctionnalité précédemment accessible depuis l'application de consultation des archives.

Le module d'archivage de Workey v.5 nécessitait obligatoirement de disposer des fichiers xml de déploiement. Avec le passage du modélisateur *GraphTalk* aux modélisateurs FLASH puis HTML5, la structure de ces fichiers a évolué à plusieurs reprises, rendant impossible de proposer une mécanique homogène et adaptative de génération des modèles (templates) *Velocity*.

C'est pourquoi une **génération automatique de modèles par défaut** a été mise en place. Cette génération est réalisée, au fil de l'eau, lors de l'archivage des documents. Pour un document donné, et **spécifiquement pour la version du processus à laquelle il appartient**, si aucun template par défaut n'existe déjà dans le répertoire templates/default/, alors un modèle est créé **à partir du Formulaire d'archivage**. vec une nomenclature spécifique.

Concaténation:

- du nom interne du type de document
- des caractères -v (trait d'union et la lettre v minuscule) de séparation
- du **numéro de version du type de processus**
- de l'extension .vml

Attention

La personnalisation des modèles *Velocity* n'est pas traitée dans cette présente documentation. Elle requiert des connaissances techniques spécifiques.

Mise-à-jour du /metadata.js

Lors de l'archivage d'un document, les métadonnées sont complétées. S'ils ne sont pas déjà présents dans le fichier, les noms internes et les libellés des types de document et champs archivés sont ajoutés aux tableaux JSON correspondants: documentTypes et fields.

Export XML du document

Le document ouvert est exporté au format xml vers un fichier placé dans le répertoire documents/ de l'archive. La sérialisation est assurée par l'API Java de Workey. Un document archivé reprend donc la structure standard avec quelques informations complémentaires.

L'élément <wky:document> est complété:

- des attributs:

- parent-id, l'identifiant numérique du document père (uniquement si le document présente un document père)
- archive-date, la date à laquelle le document a été archivée (date normalisée selon le modèle yyyy-MM-dd'T'HH:mm:ss.SSS).
- hash-dir, le chemin relatif depuis le répertoire documents/ vers le présent fichier xml.
- et le cas échéant, d'un sous élément <children-documents>, contenant lui-même autant de sous-éléments <child> ayant en attribut l'id de chacun de ses fils directs.

Exemple:

parent-id="968" archive-date="2017-06-01T15:07:28.941" hash-dir="0d/0c"

Indexation

Rappel: l'indexation est assurée par la même brique fonctionnelle que celle du moteur Workey.

En revanche, pour les besoins spécifiques au module d'archivage, sont aussi indexées les informations suivantes:

- la **date d'archivage** du document; clé: Workey_ArchiveDate
- les **droits d'accès** au document archivé:
 - **nominatifs**; désignation des acteurs par leur identifiant numérique (ACTORS.ID); clé: Workey_PRK
 - **publics**; c'est-à-dire via les noms internes des rôles (ROLES.DESIGNER_NAME); clé: Workey_PublicAccessRoleDesignerName
- la **clef de hachage** permettant d'accéder au fichier xml de données du document dans l'arborescence du répertoire documents/; clé: Workey_HashDir
- les **relations père-fils**, avec la liste des identifiants numériques des documents fils; clé: Workey_ChildDocId

Export des pièces jointes

Chacune des pièces jointes du document archivé est recopiée vers le répertoire attachments/ de l'archive, avec renommage (cf. [nomenclature](#) des fichiers de ce répertoire).

Information

Si une même pièce jointe est partagée entre plusieurs documents d'une même famille, alors cette copie aura lieu autant de fois qu'elle est référencée; le fichier existant étant écrasé par la nouvelle copie.

Export des ressources de localisation

A chaque document archivé, appartenant donc à une version précise d'un type de processus, la présence de ressources de localisation est vérifiée. Si aucune ressource n'est trouvée, alors il a création d'autant de fichiers de ressource qu'il y a de langues supportées dans la configuration du serveur.

Ces fichiers sont créés dans le répertoire locales/ avec la nomenclature détaillée [ici](#).

Rollback

L'atomicité implique l'obligation d'un retour arrière (*rollback*) en cas d'échec de l'archivage d'un document ou en cas d'interruption.

Le périmètre de ce *rollback* est défini par les documents de la famille en cours de traitement lors de la survenue de l'exception (erreur, échec, interruption, etc...). Les opérations menées en cas de *rollback* sont les suivantes:

- suppression des fichiers nouvellement créés dans les arborescences documents/ et attachments/. En revanche, il n'y aura pas suppression des répertoires intermédiaires, car ceux-ci sont potentiellement utilisés par les clefs de hachages d'autres documents déjà archivés avec succès.
- la base d'index n'est pas mise à jour avec le résultat de l'indexation des documents de la famille.

Ne sont pas concernés par le *rollback* les éléments découlant directement des métadonnées du processus:

- les modèles *Velocity* par défaut qui auraient été générés.
- les fichiers de ressources de localisation nouvellement exportés,
- les métadonnées des éléments de recherche (le fichier /metadata.js).

Suppression

Lorsque l'exportation des documents d'une famille, vers la base d'archive, est réalisée avec succès, alors ceux-ci peuvent être supprimés de la base socle du serveur Workey. Cette suppression est nécessairement hors-périmètre de l'atomicité et englobe les actions suivantes:

- suppression des tuples correspondant au(x) document(s) dans les différentes tables de la base socle. Ces suppressions sont effectuées à l'aide de requête SQL (*batchées*) ciblant les tuples par leur clef primaire.
- invalidation des caches d'*Hibernate*, pour éviter toute désynchronisation avec la base directement modifiée via JDBC.
- suppression des éventuelles pièces jointes du serveur Workey. Cette suppression repose sur la brique fonctionnelle du moteur gérant les pièces jointes (recours à la fonction de purge).
- désindexation des documents de la base Lucene du serveur Workey.

E-mail aux Gestionnaires de Workflow

En cas d'interruption du traitement d'archivage, et uniquement dans ce cas, un email d'alerte est envoyés aux seuls acteurs (de classe LDAP 'Person') disposant des droits Gestionnaire de Workflow.

Cette notification a pour sujet (localisé au regard de la langue par défaut paramétrée au niveau du serveur Workey):

En français: L'agent d'archivage Workey a rencontré une erreur.

En anglais: The Workey archiving agent has encountered an error.

Toutefois, le corps du mail est en anglais et expose les éléments techniques détaillant la nature de l'interruption, ainsi que l'état d'avancement de l'archivage au moment de sa survenue.

Exemple:

*** This message has been sent from the Workey Server running at <http://10.225.7.60:8080/workey/workey> ***

The archiving Agent, with ID 9, started at: Mon Mar 15 15:01:35 CET 2021 has unexpectedly ended at Mon Mar 15 15:04:55 CET 2021.

The following process type(s)/document(s) have been processed:

```
.process type: 'AMELIA', 64 document(s) processed.
.successfully exported/archived document(s): 64
  document ID(s): [18800, 18941, 19251, 19433, 19493, 19616, 20220, 20332, 20337, 20378, 20413, 20427, 20742, 21006, 21318,
21425, 21871, 21959, 22128, 22212, 22278, 22301, 22304, 22390, 22512, 22594, 22654, 22698, 22710, 22715, 22724, 22799, 22806,
22814, 22816, 22818, 22819, 22820, 22823, 22825, 22827, 22830, 22845, 22878, 22902, 22911, 22975, 22985, 22988, 22999, 23020,
23046, 23054, 23056, 23057, 23059, 23060, 23061, 23070, 23073, 23082, 23123, 23128, 23129]
.skipped/discarded document(s): 0

.process type: 'PROPAM', no document processed.

.process type: 'AMELIA 3', no document processed.

.process type: 'Process_bidon', no document processed.

.process type: 'AUDEXT', no document processed.

.process type: 'ANO_RECL', no document processed.
```

Failure message: Invalid archive name specified.

Root cause: Invalid archive name specified.

Full stack trace follows:

```
java.lang.Exception: Invalid archive name specified.
  at com.clog.workey.archive.Archiver.archiveDocuments(Archiver.java:414)
  at com.clog.workey.archive.Archiver.doExport(Archiver.java:370)
  at com.clog.workey.agents.tasks.ProcessArchiver.archive(ProcessArchiver.java:279)
  at com.clog.workey.agents.tasks.ProcessArchiver.process(ProcessArchiver.java:125)
  at com.clog.workey.agents.WorkeyAgent.execute(WorkeyAgent.java:226)
  at org.quartz.core.JobRunShell.run(JobRunShell.java:223)
  at org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:549)
```

Exemple de mail adressé aux Gestionnaires de Workflow.

Les éléments techniques sont les suivants:

- l'URL à laquelle le serveur est censé être joignable. Il s'agit ni plus ni moins de la valeur de la propriété `com.clog.workey.engine.ExternalURL.serverContext` de configuration du serveur.
- l'identifiant numérique de l'Agent d'archivage
- la date de déclenchement de l'instance de l'Agent
- la date d'interruption du traitement d'archivage
- la liste des processus déjà traités durant cette exécution. Pour chacun, le nom interne du processus, avec le nombre total de documents traités, ventilés le cas échéant:
 - les documents archivés avec succès (avec le tableau des identifiants numériques des documents)

- les documents ignorés (cf. [éviction des documents avortés](#))
- Si une exception Java a été jetée, alors celle-ci est explicitée:
 - Failure message : le message d'erreur remonté
 - Root cause : la cause racine, en cas d'encapsulation
 - Full stack trace follows : la pile d'exécution à l'instant de l'exception.

Application de consultation des archives

Accès à l'application

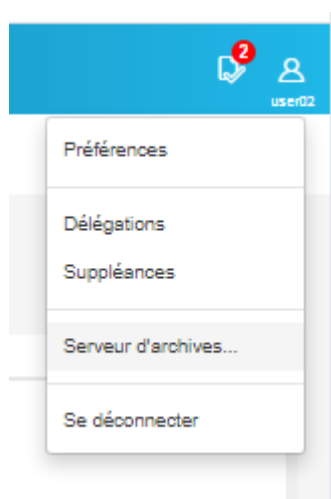
La servlet de l'application est embarquée et définie dans le même fichier `workey.war` que l'application Web de Workey. Son accès est sécurisé par le même module d'authentification. En conséquence, un utilisateur authentifié dans l'application Web, le sera automatiquement dans l'application de consultation des archives.

Information

L'accès à l'application de consultation des archives est conditionné à l'activation de la fonctionnalité d'archivage au niveau de la licence. Dans le cas contraire, l'utilisateur recevra un code d'erreur `<code>HTTP 503</code>` en retour.

Il y a deux moyens pour accéder à cette interface:

- soit directement via l'URL de la page d'accueil: `/workey/archives/home`
- soit depuis l'application Web de Workey, via le menu utilisateur. A noter que le lien ne sera proposé que si la fonctionnalité d'archivage est activé au niveau de la licence



Lien d'accès vers l'application de consultation des archives, depuis l'application Web de Workey

Information

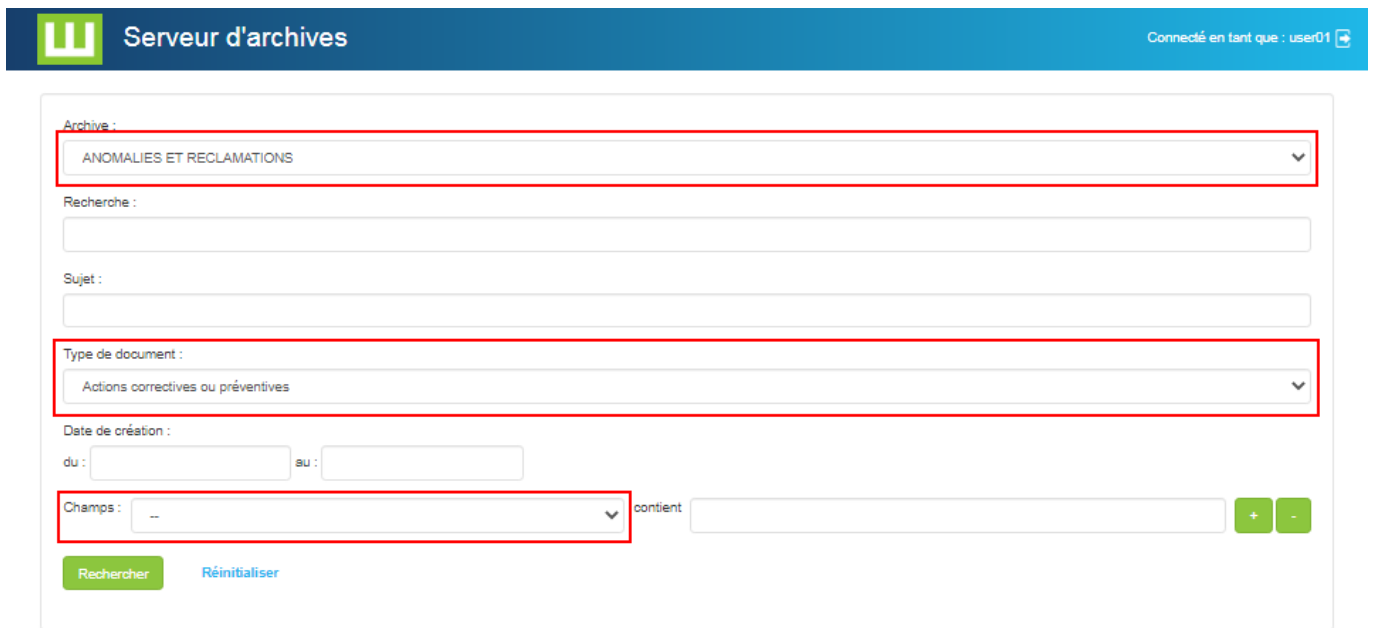
Du point de vue de Spring Security, un utilisateur authentifié dans Workey se verra attribuer le rôle `WK_ARCHIVES_ROLE` si l'archivage est activé au niveau

de la licence, en plus du rôle WORKEY_ROLE.

Page d'accueil

La page principale consiste en un formulaire de recherche avancée. L'utilisateur est invité à sélectionner la base d'archive qu'il souhaite requêter., depuis la liste déroulante alimentée avec les noms des bases d'archives présentes dans le répertoire d'archive global.

Les liste déroulantes "Type de document" et "Champs" sont peuplées avec les informations contenues dans le fichier /metadata.js de l'Archive sélectionnée.



Information

La liste de "Champs" n'est pas dépendante de la sélection réalisée dans la liste "Type de document".

Recherche d'un document archivé

En cas de recherche effectuée par un utilisateur ne disposant pas de privilèges de Gestionnaire de Workflow, l'application d'un filtre sur ses droits spécifiques – tant nominatifs, que publics du fait de ses affectations aux Rôles – sera assurée de la même manière que pour une recherche de documents dans l'application Web de Workey.

Le résultat de la recherche est trié par ordre décroissant de pertinence.

Une limitation hardcodée a été fixée à 10000 résultats.

Consultation d'un document archivé

L'utilisateur peut accéder à un document en cliquant sur son sujet depuis la

liste des résultats de sa recherche.

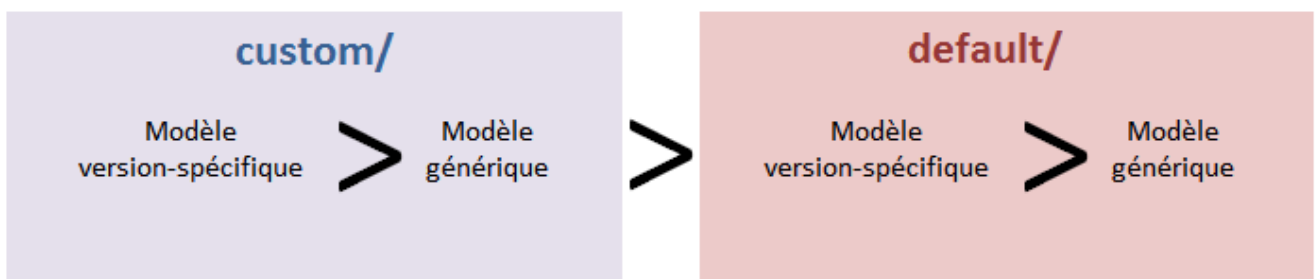
Détermination du modèle *Velocity*

Au regard du **type de document** et de la **version du processus** auquel appartient le document accédé, la servlet va sélectionner un modèle *Velocity* présent dans l'arborescence des répertoires templates/.

Parmi les modèles éligibles, l'algorithme sélectionnera un modèle en privilégiant toujours:

- un modèle version-spécifique sur un modèle générique
- un modèle personnalisé sur un modèle par défaut

Il découle de ces deux principes l'ordre préférentiel suivant:



Information

Pour les documents archivés à partir de la v.6, l'agent d'archivage génère automatiquement un modèle version-spécifique par défaut pour chaque type de document traité.

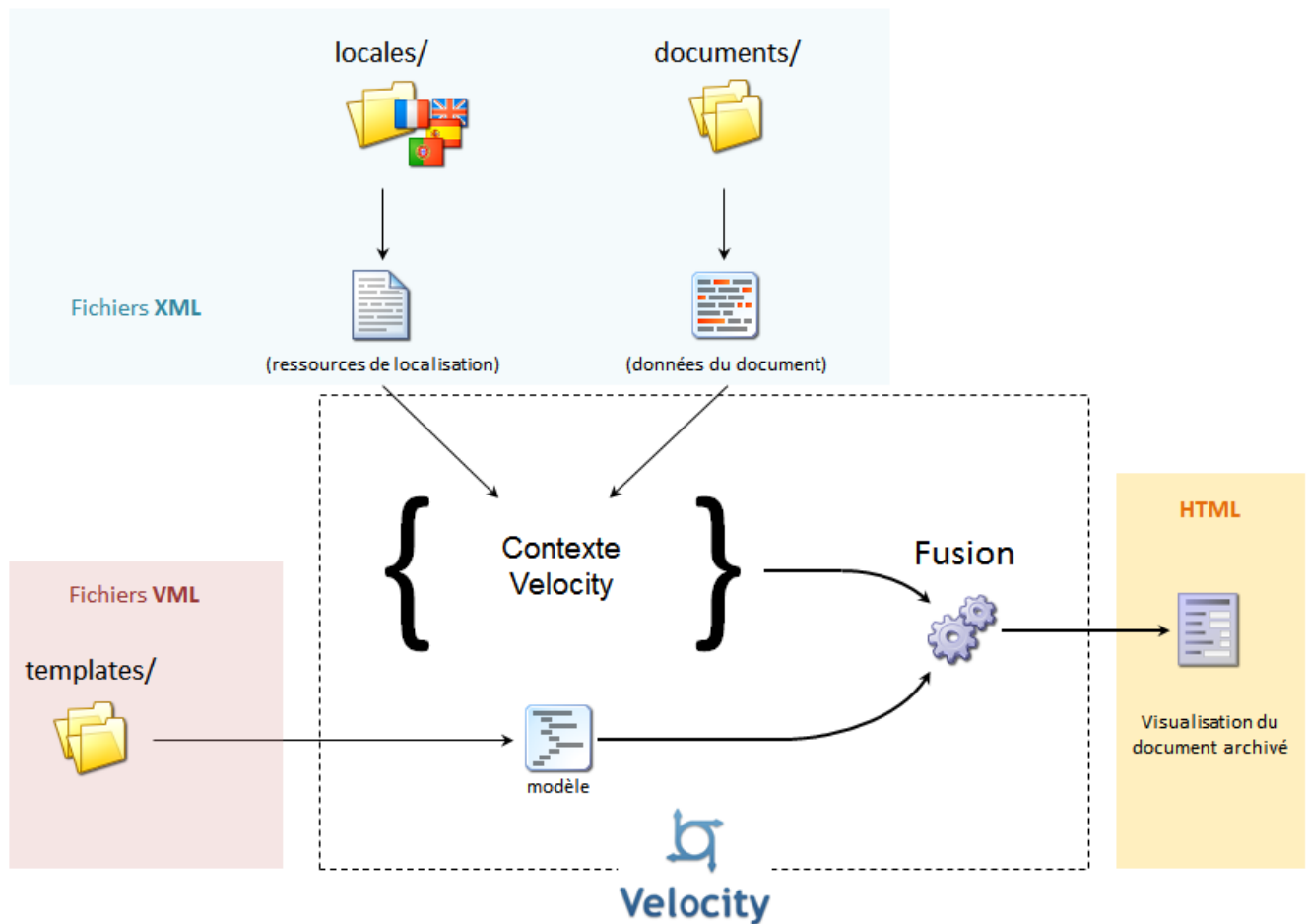
Pour les documents archivés avant la v.6, leurs modèles devront être correctement localisés dans les sous-répertoire de templates/ et répondre à la nomenclature correspondant à leur spécificité (générique ou version-spécifique).

Attention

Un modèle est **indispensable** pour générer le code HTML de la page affichant le document consulté.

Contexte de la fusion *Velocity*

Pour desservir le document consulté, l'application réalise une fusion des données du document et des ressources de localisation à l'aide d'un modèle *Velocity*.



Les données des fichiers xml sont injectées, sous forme de couples clé/objet, dans une Map (le contexte) qui sera passée en paramètre à Velocity.

Injection des ressources localisées

Au regard de la langue préférentielle de l'utilisateur (propriété `language` de l'objet `UserAuth` issu de l'authentification), le contenu du fichier de ressources correspondant est ajouté à la Map du contexte Velocity.

Chacun des éléments de localisation du fichier est ajouté à une nouvelle Map avec une clef constituée par concaténation:

- du type de la ressource
- du caractère séparateur `|` (*pipe*)
- du nom interne de la ressource

Cette objet Map, regroupant tous les éléments de localisation, est ajouté au du contexte Velocity avec la clé `langRes`.

Exemple:

```

{
  ...
  langRes: {
    document-field|demandeur: "Demandeur",
    document-field|modele: "Référence du modèle",
    ...
    operation|Creer : "Créer"
  }
}

```

```

        operation|Finir : "Finir"
        ...
        state|cree: "créé"
        ...
        role|responsable: "Responsable validation"
        ...
    }
    ...
}

```

Une exception s'applique pour les éléments des vues embarquées; notamment pour les clefs des noms localisés des colonnes. Celles-ci sont constituées par concaténation des éléments suivants:

- du type de la ressource, en l'occurrence view-column
- du caractère séparateur | (*pipe*)
- du nom interne de la vue embarquée
- du caractère séparateur | (*pipe*)
- du nom interne de la colonne

Exemple:

```

{
    ...
    view-column|vue_docs_fils_action|N__Traitement_ou_Action : "N° Traitement
ou Action"
    view-column|vue_docs_fils_action|COL_1: "Libellé",
    view-column|vue_docs_fils_action|COL_SUBJECT: "Objet de l'action",
    view-column|vue_docs_fils_action|COL_ID_DOC: "Référence de l'action",
    ...
}

```

Injection des données du document

Pour chaque champ du document archivé, un objet représentant les valeurs du champ sera associé (dans la Map du contexte) au nom interne du champ. Cet objet sera la liste ordonnée (ArrayList, potentiellement vide) des valeurs du champ.

```

{
    modele: [],
    couleurs_selectionnees: ["bleu","vert"],
    demandeur: ["Jean"],
    date_demande: ["2022-01-01T00:00:00.000"],
    ...
}

```

Les valeurs seront généralement des chaînes de caractère (String). Toutefois si le champ est de type pièce jointe, alors les valeurs de la liste seront elles-même des Map détaillant les caractéristiques de chacun des fichiers.

```

{
    ...
    demandeur: ["Jean"],

```

```

date_demande: ["2022-01-01T00:00:00.000"],
justificatifs: [
  { name: "sa25-général.pdf"
    type: "application/pdf"
    uuid: "4C0A66D8AC168804016E2194636DE7C4"
    href: "attachments/1111/4C0A66D8AC168804016E2194636DE7C4_sa25-
general.pdf"
  },
  { name: "test.pdf"
    type: "application/pdf"
    uuid: "A7BFCEd2AC168804016E2194D8FA6B99"
    href: "attachments/1111/A7BFCEd2AC168804016E2194D8FA6B99_test.pdf"
  },
  ...
]
}

```

- name, le nom original du fichier (avec son extension)
- type, le type MIME du fichier
- uuid, l'identifiant unique de la pièce jointe
- href, l'URL relative qui devra être utilisée pour télécharger le fichier de la pièce jointe. Celle-ci est constitué de la concaténation des éléments suivants:
 - la chaîne attachments/
 - l'identifiant numérique du document suivi d'un /
 - l'uuid du document
 - un caractère `_` (*underscore*) séparateur
 - le nom normalisé du fichier (cf. [normalisation du nom du fichier original](#))

Information

Les valeurs des champs de type date sont sérialisées par l'API Workey au format standardisé [ISO 8601](#) : yyyy-MM-dd'T'HH:mm:ss.SSS

Injection des données des vues embarquées

Le contenu des vues embarquées est archivée sous forme matricielle (une liste de liste de valeurs textuelles). Chaque vue sera associée, par le biais de son nom interne, à la liste ordonnées (ArrayList, potentiellement vide) des lignes qui la constitue; chaque ligne étant elle même un objet (une Map) dans lequel les valeurs textuelles sont associée au nom interne de leur colonne.

Exemple:

```

{  vue_docs_fils_action: [
    { COL_1: "",
      COL_SUBJECT: "Fils n°1253545 pour action",
      COL_ID_DOC: "123545",
      Workey_Id: "123545"
      Workey_Subject: "Fils n°1253545 pour action"
    },
  ]
}

```

```

        {
            COL_1: "",
            COL_SUBJECT: "Fils n°127426 pour action",
            COL_ID_DOC: "127426",
            Workey_Id: "127426"
            Workey_Subject: "Fils n°127426 pour action"
        },
        ...
    ],
    ...
}

```

A noter que chaque ligne d'une vue correspond à un document Workey; c'est pourquoi l'objet représentant une ligne contient, en plus des valeurs des colonnes, deux informations complémentaires:

- Workey_Id, l'identifiant interne du document Workey pour cette ligne
- Workey_Subject, le sujet du document Workey pour cette ligne

Injection de l'historique du document

Le détail de l'historique du document est ajouté à la Map du contexte sous la forme d'une liste ordonnée (ArrayList, théoriquement jamais vide) associé à la clé history. Chaque valeur de cette liste est un objet (une Map) contenant le détail d'une entrée d'historique.

Exemple:

```

{
    ...
    history: [
        {
            isoDate: "2020-12-07T10:53:38.030",
            date: @java.Util.Date,
            actor: "Jean",
            operation: "Creer",
            state: "Cree"
        },
        {
            isoDate: "2020-12-07T10:54:24.530",
            date: @java.Util.Date,
            actor: "Jean",
            operation: "Finir",
            state: "Termine"
        },
        ...
    ]
}

```

Les propriétés (couples clé/valeur) de ces entrées d'historique sont les suivantes. Sauf exception, les valeurs sont des chaînes de caractère (String)

- isoDate, la date de réalisation de l'opération, normalisé ISO 8601
- date, une instance de java.util.Date représentant la même date que isoDate
- actor, le nom de l'acteur ayant réalisé l'opération

- operation, le nom interne de l'opération réalisé
- state, le nom interne de l'état atteint à l'issue de l'opération

Injection des informations générales

En complément, le contexte est peuplé d'un ensemble d'informations communes aux documents Workey.

- Workey_Id, l'identifiant numérique du document
- Workey_Subject, le sujet du document
- Workey_Process, le nom interne du type de processus auquel appartient ce document
- Workey_DocType, le nom interne du type de document dont relève ce document
- Workey_State, le nom interne de l'état final du document
- Workey_ParentId, l'identifiant numérique de l'éventuel document père (peut être null)
- Workey_ArchiveDate, la date d'archivage du document (formatée ISO 8601)
- Workey_ChildrenDocuments, uniquement si le document présente des documents fils. La valeur associée est alors la liste (ArrayList) des identifiants numériques des documents fils (String)

Exemple:

```
{
  ...
  Workey_Id: "123544",
  Workey_Subject: "Doc N°123544",
  Workey_Process: "Processus_de_test",
  Workey_DocType: "Mon_type_de_document"
  Workey_State: "Termine",
  Workey_ParentId: null,
  Workey_ArchiveDate: "2021-02-22T17:10:13.399"
  Workey_ChildrenDocuments: ["123545","127426"],
  ...
}
```

Limitations de la visualisation des documents archivés

Aux différentes limitations (du [module d'archivage](#) ou de [la modélisation](#)) s'ajoutent les limitations suivantes, spécifiques à la visualisation des documents archivés.

Les listes de valeur des domaines associés aux champs ne sont pas disponibles une fois le document archivé. Par conséquent, toutes les apparences liées à un choix (multiple ou non), telles que les case-à-cocher, les ensembles de boutons radio, les listes déroulante, etc. ne seront pas prises en charge. Seule la ou les valeurs sélectionnées seront affichées.

Certaines mises en forme définies dans le formulaire d'archivage ne sont pas conservées. C'est notamment le cas des champs disposés dans des tables statiques; leur ventilation par colonne (ou par césure en v.5) n'est pas reprise dans les modèles par défaut générés par l'agent d'archivage.

À éviter

Les données exposées par le formulaire d'archivage seront visibles par l'ensemble des utilisateurs ayant accès au document archivé !

Si du code javascript était utilisé pour contrôler la visibilité de certains éléments du formulaire (masquage de champs ou de sections), alors celui-ci sera caduque car inopérant lors de la consultation des documents archivés !

Modèles *Velocity*

La création ou la modification de modèle *Velocity* ne sera pas traitée dans le cadre de cette documentation. Pour de plus amples informations:

<https://velocity.apache.org/>

Macros spécifiques

Afin de faciliter l'exploitation et la restitution des données archivées injectées dans le contexte *Velocity*, un ensemble de macros ont été définies. Les modèles générés automatiquement par l'agent d'archivage s'appuient sur ces macros. Pour plus de détail sur l'implémentation de ces macros, se référer au fichier `VM_archive.vm` situé à l'intérieur de l'archive web `workey.war` sous le chemin: `/WEB-INF/classes/com/workey/archiver/templates/`

Macro (paramètres...)	Description
<code>#label(\$langKey, \$defaultLabel)</code>	Retourne l'élément de localisation correspondant à la clé <code>\$langKey</code> passée en premier paramètre. Si aucune localisation n'est associée à cette clé dans la map <code>langRes</code> du contexte, alors retourne le libellé <code>\$defaultLabel</code> passé en second paramètre.
<code>#fieldValue(\$field, \$index)</code>	Retourne la valeur, à la position <code>\$index</code> , du champ désigné par <code>\$field</code> . La première valeur est pour position d'index 0 (zéro). Ne retourne rien, si le champ ne contient aucune valeur. Retourne la première valeur si le paramètre <code>\$index</code> est omis. Retourne - si la position d'index demandée excède le nombre de valeur du champ.
<code>#fieldMultiValue(\$field)</code>	Retourne l'ensemble des valeurs du champ désigné par <code>\$field</code> , sous la forme d'une <code>UnsortedList</code> HTML (élément <code></code>). Chaque valeur étant un item de liste (élément <code></code>). Ne retourne rien, si le champ ne contient aucune valeur.

Macro (paramètres...)	Description
	Retourne les lignes d'un tableau dynamique. Le paramètre obligatoire, <code>\$columnNames</code>, est un tableau contenant les noms des champs constituant le tableau dynamique. Pour chaque ligne du tableau, un élément <code><tr></code> est créé, avec autant de cellules (<code><td></code>) qu'il y a d'éléments dans <code>\$columnNames</code>.
<code>#dynamicTable(\$columnNames)</code>	Il est impératif que tous les champs du tableau aient le même nombre de valeur. Sachant que le nombre de valeur du premier des champs détermine le nombre de ligne total du tableau. Note: cette macro fait elle même appel à la macro <code>#fieldValue</code>. Son usage doit se faire au sein d'un élément <code><table></code> ou <code><tbody></code>. Retourne un lien relatif, sous la forme d'un élément <code><a></code>, d'accès à la pièce jointe à la position <code>\$index</code> du champ désigné par <code>\$field</code>. Le contenu de l'élément correspond au nom original du fichier. L'attribut <code>href</code> est renseigné avec la valeur correspondant à la clé <code>href</code> dans le contexte <i>Velocity</i>. Contrairement à la macro <code>#fieldValue</code>, l'appelant doit s'assurer que la position d'index n'excède pas le nombre de valeur du champ.
<code>#linkAttachment(\$field, \$index)</code>	
<code>#linkAttachmentMV(\$field)</code>	Retourne l'ensemble des liens d'accès aux pièces jointes du champ désigné par <code>\$field</code>, sous la forme d'une <code>UnsortedList</code> HTML (élément <code></code>). Chaque élément <code><a></code> étant encapsulé dans un item de liste (élément <code></code>). Ne retourne rien, si le champ ne contient aucune valeur. Note: cette macro fait elle même appel à la macro <code>#linkAttachment</code>.
<code>#viewContent(\$view, \$columnNames)</code>	Retourne les lignes de la vue embarquée désignée par <code>\$view</code>. Le paramètre obligatoire, <code>\$columnNames</code>, est un tableau contenant les noms internes des colonnes constituant la vue. Pour chaque ligne de la vue, un élément <code><tr></code> est créé, avec autant de cellules (<code><td></code>) qu'il y a d'éléments dans <code>\$columnNames</code>. Note: son usage doit se faire au sein d'un élément <code><table></code> ou <code><tbody></code>.
<code>#isEmpty(\$field)</code>	Retourne <code>false</code> si le champ <code>\$field</code> existe et qu'il contient au moins une valeur et que la première valeur est non vide. Retourne <code>true</code> dans le cas contraire.
<code>#formatDate(\$field, \$index)</code>	Retourne la date, à la position <code>\$index</code>, du champ désigné par <code>\$field</code>. La première date à pour position d'index 0 (zéro). La date est formatée de sorte à ne retenir que la partie calendaire (pas d'information d'heure). Le formatage est localisé par rapport à la langue préférentielle de l'utilisateur. Le pattern utilisé correspond à celui défini par <code>java.text.DateFormat.LONG</code>. Note: macro destinée à être utilisée avec les valeurs des champs de type date (qui ont été sérialisées/normalisées ISO 8601 lors de l'archivage).

Macro (paramètres...)	Description
#formatDateTime(\$string)	Retourne la date et l'heure reformatés, à partir d'une date sérialisée/normalisée ISO 8601 Le formatage est localisé par rapport à la langue préférentielle de l'utilisateur. Le pattern utilisé correspond à celui défini par <code>java.text.DateFormat.MEDIUM</code> à la fois pour la date et pour l'heure. Note: macro destiné à être utilisé avec les valeurs des champs de type date (qui ont été sérialisées/normalisées ISO 8601 lors de l'archivage).
#documentHistory	Retourne les lignes du tableau d'historique du document archivé. Chaque entrée d'historique est traduite en élément <code><tr></code>, avec des éléments <code><td></code> pour: .la date et l'heure (<code>#formatDateTime</code>), .l'acteur ayant réalisé l'opération, .le nom localisé de l'opération, .le nom localisé de l'état atteint. Se charge d'ajouter, en dernière entrée, l'opération d'archivage. Note: son usage doit se faire au sein d'un élément <code><table></code> ou <code><tbody></code>.
#startHTML	Insère en tant qu'entête le template interne <code>/WEB-INF/classes/com/workey/archiver/templates/header.vml</code> situé à l'intérieur de l'archive web <code>workey.war</code>
#endHTML	Insère en tant que bas de page le template interne: <code>/WEB-INF/classes/com/workey/archiver/templates/footer.vml</code> situé à l'intérieur de l'archive web <code>workey.war</code>

Debug

Packages et classes java pouvant être tracés dans les logs:

- pour l'agent d'archivage (celui-ci fait appel aux fonctionnalités noyau d'archivage): `com.clog.workey.agents.tasks.ProcessArchiver`
- pour les fonctionnalités noyau d'archivage: `com.clog.workey.archive`
- pour l'application de consultation des archives: `com.workey.archiver`
- pour la fusion *Velocity*: `com.workey.archiver.Velocity`