

Propriétés serveur Workey

Les propriétés détaillées dans cette section permettent de configurer les différentes parties fonctionnelles du serveur Workey.

Elles doivent être présentes dans le fichier `<CATALINA_HOME>/conf/catalina.properties`.

Ce fichier définit un ensemble de propriétés (des couples clef – valeur) qui seront ajoutées aux propriétés système Java au lancement de *Tomcat*.

Cette liste de propriétés sera chargée à l'aide de la méthode *load* de la classe `java.util.Properties`.

Les lignes du fichier commençant par **#** sont considérées comme des **commentaires**.

Les espaces précédant une valeur sont ignorés. En revanche la chaîne de caractère qui est retenue comme valeur s'étend du premier caractère (autre qu'un espace) jusqu'à la fin de ligne (les espaces après la valeur sont conservés).

Si une URL doit être spécifiée, elle devra aussi respecter la norme **RFC 3986**. Exemple : échappement du caractère d'espacement à l'aide de **%20**

Si un chemin sur le système de fichier doit être spécifié, le caractère **/** devra être utilisé comme séparateur de répertoire. Il est toutefois possible d'utiliser des **** mais il sera alors nécessaire de les échapper par un autre ****.

Exemple :

`C:\\Tomcat\\workey\\my_folder`

ou

`..\\workey\\my_folder`

Mise en garde sur l'utilisation de chemins relatifs.

-> selon que le serveur Workey est lancé en tant que service ou en console, le répertoire d'exécution (spécifique à l'identité utilisée) est susceptible d'être différent. Ceci pourrait potentiellement fausser la résolution des chemins relatifs. Les fichiers de propriétés java standards **ne** supportent **pas** la substitution de variables.

Journalisation

Pour la journalisation, le serveur Workey utilise le framework *Log4J*. Celui-ci embarque, dans sa librairie, son propre fichier de configuration par défaut.

La configuration de celui-ci se fait par le biais d'un fichier de propriété spécifique. Pour spécifier une configuration différente, il est nécessaire de lui indiquer, à l'aide de la propriété suivante, le chemin vers un autre fichier de configuration :

- **log4j.configuration**

Pour renseigner cette propriété avec le chemin d'un fichier de configuration alternatif, il est nécessaire de préciser que ce chemin est accessible via le système de fichier (en préfixant le chemin avec **file:**)

Dans le cas contraire, [le fichier sera d'abord recherché sur le classpath du classloader](#).

Exemple: log4j.configuration=file:../conf/log4j.properties

De nombreuses ressources sont disponibles sur Internet pour vous aider dans la configuration de ce framework de journalisation (cf. [Annexe E – Ressources externes](#)).

Répertoires

Au démarrage, le serveur Workey vérifie qu'il dispose d'un ensemble de chemins vers ses répertoires de travail.

Ci-contre, l'arborescence des répertoires qui seront créés au démarrage du serveur Workey.

La mise en place de cette arborescence découle du comportement suivant:

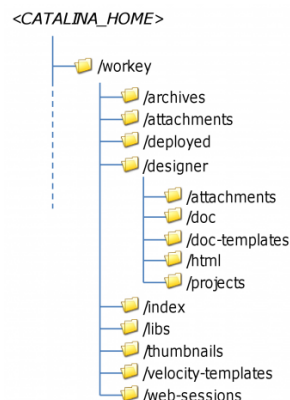
– Lorsqu'un répertoire n'est pas présent, et que la **propriété correspondante n'a pas été renseignée** :

-> alors il sera créé au regard du chemin par défaut.

– En revanche, tout chemin **expressément renseigné** devra pointer vers un répertoire **existant et accessible** :

-> celui-ci ne sera pas créé.

Ci-après, la liste des propriétés permettant de stipuler les emplacements des différents répertoires de travail du serveur Workey.



Arborescence de Workey
(à partir de 6.2.0)

Propriété	Description	Chemin par défaut
com.clog.workey.directory	-> Répertoire racine de Workey. Il sert de base à l'arborescence par défaut des sous-répertoires dédiés aux diverses parties fonctionnelles de Workey. Le fichier de licence workey-license.xml doit être placé à la racine de ce répertoire.	Création d'un répertoire /workey sous le chemin désigné par la propriété java catalogina.home . Cette propriété java est renseignée avec le chemin contenu dans la variable d'environnement <CATALINA_HOME> qui désigne le répertoire d'installation de Tomcat .
com.clog.workey.directory.archives	-> Répertoire d'export des documents Workey archivés.	Création d'un répertoire /archives sous le <i>répertoire racine de Workey</i> .
com.clog.workey.directory.attachments	-> Répertoire de stockage des pièces jointes des documents Workey.	Création d'un répertoire /attachments sous le <i>répertoire racine de Workey</i> .
com.clog.workey.directory.deployed	-> Répertoire de stockage des XML des processus déployés.	Création d'un répertoire /deployed sous le <i>répertoire racine de Workey</i> .
com.clog.workey.directory.designer	-> Répertoire racine du modélisateur Workey.	Création d'un répertoire /designer sous le <i>répertoire racine de Workey</i> .
com.clog.workey.directory.designer.attachments	-> Répertoire de stockage des fichiers joints aux projets modélisés.	Création d'un répertoire /attachments sous le <i>répertoire racine du modélisateur</i> .
com.clog.workey.directory.designer.docTemplates (Cette propriété n'est disponible qu'à partir de Workey 6.2.0)	-> Répertoire de stockage des fichiers de modèles pour la génération de la documentation au format Word.	Création d'un répertoire /doc-templates sous le <i>répertoire racine du modélisateur</i> .
com.clog.workey.directory.designer.genDoc	-> Répertoire temporaire de génération de documentation au format Word.	Création d'un répertoire /doc sous le <i>répertoire racine du modélisateur</i> .

Propriété	Description	Chemin par défaut
<code>com.clog.workey.directory.designer.genHTML</code>	-> Répertoire <i>temporaire</i> de génération de documentation au format HTML	Création d'un répertoire /html sous le <i>répertoire racine</i> du <i>modélisateur</i> .
<code>com.clog.workey.directory.designer.projects</code>	-> Répertoire de <i>stockage</i> des projets modélisés .	Création d'un répertoire /projects sous le <i>répertoire racine</i> du <i>modélisateur</i> .
<code>com.clog.workey.directory.index</code>	-> Répertoire de <i>stockage</i> des index Lucene de Workey.	Création d'un répertoire /index sous le <i>répertoire racine</i> de <i>Workey</i> .
<code>com.clog.workey.directory.libs</code>	-> Répertoire de librairies supplémentaires .	Création d'un répertoire /libs sous le <i>répertoire racine</i> de <i>Workey</i> .
<code>com.clog.workey.directory.scripts</code> (Cette propriété n'est disponible qu'à partir de Workey 6.4.0)	-> Répertoire contenant les scripts utilisateurs (utilisé notamment par le connecteur ScriptConnector).	Création d'un répertoire /scripts sous le <i>répertoire racine</i> de <i>Workey</i> .
<code>com.clog.workey.directory.thumbnails</code>	-> Répertoire de <i>stockage</i> des vignettes (<i>thumbnails</i>) des pièces jointes de documents Workey.	Création d'un répertoire /thumbnails sous le <i>répertoire racine</i> de <i>Workey</i> .
<code>com.clog.workey.directory.velocity-templates</code>	-> Répertoire de <i>stockage</i> des templates velocity utilisés pour les besoins de fusion HTML .	Création d'un répertoire /velocity-templates sous le <i>répertoire racine</i> de <i>Workey</i> .
<code>com.clog.workey.directory.web-sessions</code>	-> Répertoire utilisé pour la création de fichiers temporaires liés aux sessions des utilisateurs de l'Application Web de Workey.	Création d'un répertoire /web-sessions sous le <i>répertoire racine</i> de <i>Workey</i> .

NE JAMAIS UTILISER UN MÊME RÉPERTOIRE DE STOCKAGE DES PIÈCES JOINTES ENTRE DIFFÉRENTES INSTANCES DE WORKEY.

D'autre part, il doit être distinct du répertoire de stockage des vignettes.

NE JAMAIS UTILISER UN MÊME RÉPERTOIRE DE STOCKAGE DES VIGNETTES ENTRE DIFFÉRENTES INSTANCES DE WORKEY.

D'autre part, il doit être distinct du répertoire de stockage des pièces jointes.

Annuaire LDAP

Avant d'aborder la section suivante, il est nécessaire de comprendre l'interaction entre le serveur Workey et les différents annuaires.-> Pour cela, vous pouvez vous reporter à l'[Annexe B – Annuaire LDAP](#).

Propriétés de connexion aux annuaires

- Rappel :

-> Les propriétés **com.clog.login.*** permettent de décrire l'annuaire utilisé pour la phase d'authentification des utilisateurs.-> Les propriétés **com.clog.userDirectory.*** permettent de décrire l'annuaire contenant les informations relatives aux utilisateurs authentifiés.

- Ces ensembles de propriétés doivent être définis.

Toutefois, si votre environnement n'utilise qu'un seul annuaire, alors ces deux jeux de propriétés devront être renseignés de manière à être en phase et désigner ainsi votre annuaire unique.

Propriété d'annuaire (respectivement d'authentification ou informationnel)	Description	Remarque
com.clog.login.providerUrl com.clog.workey.userDirectory.providerUrl	-> URL de connexion à l'annuaire. La syntaxe est la suivante: protocole://adresseIP/DN_racine ou protocole://nom_de_machine:port/DN_racine où DN_racine est le chemin, depuis la racine de l'annuaire, du point d'entrée dans l'arborescence. Il définit ainsi le <i>contexte racine</i>. Exemple : ldap://127.0.0.1:389/dc=c-log,dc=com	Si ce paramètre est absent ou laissé vide, Workey utilisera l'URL : ldap://localhost:389 Workey ne pourra voir que la sous-arborescence à partir de ce point d'entrée. Il en résultera que les utilisateurs situés en amont de cette arborescence ne pourront pas être vus. Il en sera de même pour les utilisateurs qui se trouveraient dans des sous- arborescences qui ne seraient pas en aval du point d'entrée.-> <i>Le contexte racine doit impérativement être spécifié dans l'URL de connexion.</i>
com.clog.login.workeyPrincipal com.clog.workey.userDirectory.workeyPrincipal	-> L'identité utilisée par le serveur Workey pour établir les connexions à l'annuaire. C'est-à-dire le compte utilisateur (aussi désigné par le terme principal). Exemple : <i>cn=admin,dc=c-log,dc=com</i>	
com.clog.login.workeyCredentials com.clog.workey.userDirectory.workeyCredentials	-> Le mot de passe associé au principal utilisé par le serveur Workey pour établir les connexions à l'annuaire.	Les connexions sont établies avec des mots de passe en clair. -> Il est donc indispensable que le serveur Workey et l'annuaire LDAP soient sur un réseau sécurisé.
com.clog.login.useTLS (optionnel) com.clog.workey.userDirectory.useTLS (optionnel)	-> Utilisation de la négociation TLS lors des connexions à l'annuaire.	Pour activer la négociation TLS, la propriété doit être présente et égale à la chaîne 'true'.

Paramétrage de la « clef utilisateur »

A titre de rappel, un annuaire respectant le modèle X.500, interrogé via le protocole LDAP, est structuré comme suit:une entrée est constituée d'un ensemble d'attributsun attribut est nommé et a une (ou plusieurs) valeur(s)chaque entrée a un identifiant unique: son **DN** (Distinguished Name). Celui-ci est constitué de son **DN relatif**, construit depuis un ou plusieurs de ses attributs, suivi du DN de son parent.

Du fait que les utilisateurs soient des entrées de l'annuaire, le choix de la "clef utilisateur" est primordial.

Gardez à l'esprit que le DN peut changer pendant la durée de vie d'une entrée, par exemple, quand les entrées sont déplacées dans l'arborescence de l'annuaire.

Afin d'identifier les entrées de manière fiable et non ambiguë, un identifiant **unique** et **invariant** pourrait faire partie des attributs de chaque entrée.

Une fois ces considérations prises en compte, il faut indiquer pour le(s) annuaire(s) si la clef des utilisateurs sera leur DN ou bien la valeur de l'un de leurs attributs.

**Propriété d'annuaire
(respectivement d'authentification ou
informationnel)**

Description

Remarque

com.clog.login.userKeyAttribute com.clog.workey.userDirectory.userKeyAttribute	-> Spécifie le nom de l'attribut LDAP contenant la clef à utiliser pour l'identification des utilisateurs.	Il est impératif que la valeur récupérée de l'annuaire , pour l'attribut LDAP spécifié, soit :clef : à défaut, ce sera le premier élément correspondant qui sera retenu (l'ordre des résultats ne pouvant pas être garanti par Workey).invariante.
	-> Si ce paramètre est absent ou laissé vide , alors les utilisateurs seront identifiés/retrouvés par leur chemin (relatif) à partir du point d'entrée dans le LDAP (c'est-à-dire, par rapport au contexte racine).	

Propriétés spécifiques à l'annuaire d'authentification

Propriété d'annuaire	Description	Remarque
com.clog.login.aliasNameAttribute	-> Permet de spécifier le nom de l'attribut contenant le login de l'utilisateur.	Pour un ADS (Active Directory Service), le login de session Windows correspond à l'attribut sAMAccountName.
	-> S'il est absent ou laissé vide, Workey tentera d'utiliser l'attribut 'cn'.	-> Pour MSeXchange 5.5, il conviendra d'utiliser l'attribut rdn.
com.clog.login.userPrincipalAttribute (optionnel)	-> Cette option concerne la seconde phase de l'authentification: la validation du mot de passe. Il indique à Workey l' attribut qui devra être récupéré afin de disposer du "principal" nécessaire à la seconde connexion OpenLDAP, car ce sera alors au LDAP : celle qui validera le couple "principal" – mot de passe.	-> Cette option se doit d'être laissée vide pour un le DN absolu qui sera utilisé pour tenter la connexion.
	-> Si cette propriété est absente ou laissée vide , Workey utilisera le DN absolu (chemin depuis la racine de l'annuaire) de l'utilisateur identifié par son login.	-> En revanche, pour ADS il est conseillé d'utiliser l'attribut distinguishedName.

Propriétés spécifiques à l'annuaire informationnel

Propriété d'annuaire	Description	Remarque
com.clog.workey.userDirectory.emailAttribute (optionnel)	-> Désigne l' attribut LDAP contenant l'adresse email des utilisateurs.	-> En l'absence de paramétrage valide de cette propriété, l'attribut utilisé sera: mail
com.clog.workey.userDirectory.displayNameAttribute (optionnel)	-> Désigne l' attribut LDAP contenant le nom complet de l'utilisateur affichable dans l'interface Workey.	-> En l'absence de paramétrage valide de cette propriété, l'attribut utilisé sera: cn

Propriété d'annuaire	Description	Remarque
com.clog.workey.userDirectory. userFilter (optionnel)	-> Cette propriété définit le filtre LDAP à appliquer pour identifier, dans l'arborescence de l'annuaire informationnel, les noeuds correspondants à des utilisateurs. -> Les annuaires de type ADS (Active Directory Service) peuvent stocker des objets binaires de très grande taille, susceptibles d'excéder la taille que peut gérer l'API LDAP utilisée par Workey. Il est donc important de spécifier un filtre efficace pour ne pas tenir compte de ces types d'objet. -> Les adresses e-mail et les noms complets des utilisateurs, contenus dans l'annuaire informationnel et accédés par Workey, sont mis en cache. Par défaut, l'âge maximum d'une entrée mise en cache est de 600 secondes. Il est possible de spécifier une période différente en indiquant un nombre (entier positif) de seconde pour cette propriété. <i>Une valeur négative entraînera une durée de validité perpétuelle pour les entrées mise en cache. Une valeur nulle reviendra à indiquer à Workey d'ignorer la valeur mise en cache et de systématiquement réinterroger l'annuaire. Un tel paramétrage doit exclusivement être réservé à des fins de test.</i> -> Permet de spécifier le timeout d'interrogation de l'annuaire (en millisecondes). -> Ce nombre entier doit être plus grand que 0 (zéro). Un entier négatif ou nul signifie qu'il n'y aura aucun timeout. Toute requête LDAP attendra indéfiniment une réponse.	-> En l'absence de paramétrage valide de cette propriété, le filtre utilisé sera : (&(objectclass=person)(objectclass=user)) -> Par le biais de la console d'administration, il est possible de spécifier un autre filtre pour rechercher directement les utilisateurs. -> Afin de renseigner la propriété avec un filtre adéquat, il est recommandé de tester ce dernier à l'aide d'un client LDAP (par exemple le programme LDAP Browser).
com.clog.workey.userDirectory. cacheMaxAge (optionnel)	<i>Les caches peuvent être vidés manuellement via la console d'administration.</i>	
com.clog.workey.userDirectory. connectTimeout (optionnel)		-> En l'absence de timeout explicitement spécifié, ce sera le timeout du réseau sous-jacent qui interrompera les requêtes.

Notification Email

La configuration de l'envoi d'email par le serveur Workey est scindée en deux parties distinctes:

- la configuration du gestionnaire de sessions email,
- la configuration des notifications email Workey.

Gestion des sessions email

Une session mail est créée à chaque envoi d'email par le serveur Workey. L'obtention et la gestion de cette session sont délégués à l'API javamail. Les paramètres de création de session seront fournis tels quels à l'api. Ces paramètres doivent être explicités sous forme de propriétés java (spécifiques à l'API javamail) à définir dans le fichier **catalina.properties**. L'ensemble des propriétés supportées sont décrites dans la [documentation officielle de javamail](#).

Le serveur Workey utilise l'API javamail uniquement pour de l'envoi d'email. Il n'est donc pas nécessaire de configurer les éléments relatifs à la récupération d'email.

Exemple : pour se connecter à un serveur SMTP standard, l'ensemble suivant de propriétés doit être défini :

```
mail.transport.protocol=smtp
mail.smtp.host=my.smtp.server
mail.smtp.port=25
mail.user=user@my.domain.com
mail.password=secret
```

Note : le protocole de transport utilisé, par défaut, pour l'envoi d'email est **smtp** (spécifié par la propriété **mail.transport.protocol**).

A partir de Workey 6.5.1, il est vivement recommandé de mettre-à-jour le paramétrage des envois d'email, du fait de l'obsolescence de certaines propriétés. Cf. la section ci-après.

Sessions mail avec authentification

Depuis la version 1.4.4 de JavaMail, il n'est plus possible de spécifier, en standard, les informations d'authentification par le biais des propriétés **mail.***.

Toutefois, l'authentification pour l'envoi d'email, dans Workey, peut être paramétrée de la manière suivante.

- Indiquer qu'une authentification est requise en renseignant à **true** la propriété **mail.<protocol>.auth**

<protocol> étant le protocole utilisé pour l'envoi des emails.

- Spécifier les nom d'utilisateur et mot de passe, respectivement à l'aide des propriétés:

-> **mail.<protocol>.user**-> **mail.<protocol>.password**Exemple :

```
mail.transport.protocol=smtp
mail.smtp.auth=true
mail.smtp.user=user@my.domain.com
mail.smtp.password=secret
```

Note : en l'absence des propriétés **mail.<protocol>.user** et **mail.<protocol>.password**, les identifiants seront récupérés depuis les propriétés obsolètes **mail.user** et **mail.password**.

Configuration de la génération des emails

Lorsqu'un email de notification est envoyé par le serveur Workey, les nom et

adresse de l'émetteur doivent être spécifiés.

Dans l'exemple suivant, le serveur à renseigné les informations de l'expéditeur avec les informations relative à l'Acteur Workey à l'origine de la notification.

Exemple :

From: "Eric FRANCK" <ef@c-log.com>

'Eric FRANCK', le *personal*, correspondra au nom affichable de l'utilisateur-> récupéré depuis l'annuaire via l'attribut spécifié par la propriété [com.clog.workey.userDirectory.displayNameAttribute](#) 'ef@c-log.com' correspondra à l'adresse e-mail de l'utilisateur-> récupérée depuis l'annuaire via l'attribut spécifié par la propriété [com.clog.workey.userDirectory.emailAttribute](#)

Les propriétés suivantes permettent de paramétrer les informations sur l'expéditeur.

Les emails envoyés sont construits sous forme de [message Mime multi-partie](#). Il est possible de préciser l'encodage utilisé pour le sujet, le corps du message et les éventuelles pièces jointes.

Propriété	Description	Remarque
<code>com.clog.workey.engine.NotificationSender.senderIdentity</code>	-> Indique l' identité qui sera utilisée pour renseigner l'entête de l'expéditeur. Deux valeurs sont possibles : SERVER : le moteur Workey utilisera son identité pour indiquer l'expéditeur dans les entêtes des e-mails. ACTOR : le moteur Workey utilisera l'identité de l'acteur qui est à l'origine de l'envoi de mail (tel qu'une notification sur une opération).	-> Si senderIdentity n'est pas renseigné (ou bien renseigné avec une valeur autre que SERVER ou ACTOR), alors le moteur considèrera qu'il doit être en mode SERVER .
<code>com.clog.workey.engine.NotificationSender.mailerName</code> (optionnel)	-> Spécifie le le nom complet qui sera indiqué en tant qu'expéditeur lorsque le senderIdentity est SERVER .	-> Valeur par défaut < WORKEY SERVER >
<code>com.clog.workey.engine.NotificationSender.mailerAddress</code> (optionnel)	-> Spécifie l' adresse e-mail qui sera indiquée en tant qu'expéditeur lorsque le senderIdentity est SERVER .	-> Il n'y a pas de valeur par défaut pour cette propriété. Elle devra être impérativement renseignée. Exemple : noreply@mydomain.com
<code>com.clog.workey.engine.NotificationSender.mailCharset</code> (optionnel)	-> Spécifie l' encodage du sujet et du corps de la notification email.	-> Valeur par défaut : utf-8
<code>com.clog.workey.engine.NotificationSender.attachmentEncoding</code> (optionnel)	-> Spécifie l' encodage des pièces jointes des notifications email.	-> Valeur par défaut : base64

Contexte des URLs (générées par Workey)

Les e-mails de notification sont susceptibles de contenir des URL qui désignent des documents Workey. Ces URL pointent sur l'application Web de Workey. Toutefois, le destinataire du mail n'est pas forcément sur l'intranet. En conséquence, l'URL doit être construite au regard du chemin d'accès que devra emprunter le destinataire pour atteindre le document. Le moteur ne pouvant pas savoir qu'il y a une indirection, il est nécessaire de le lui indiquer en précisant le contexte sous lequel est accessible l'application Web.

- `com.clog.workey.engine.ExternalURL.serverContext` (optionnel)

Concrètement, le `serverContext` correspond au début de l'URL. La partie restante de l'URL (correspondant alors au chemin relatif depuis la racine de l'application Web) sera calculée et concaténée au `serverContext`.

Exemple : `com.clog.workey.engine.ExternalURL.serverContext=http://mydomain:12345/myContext/workey`

Si la propriété est absente ou laissée vide, un contexte par défaut sera utilisé : <http://localhost:8080/workey/workey>

Attention, quel que soit le contexte, il doit se terminer par : `/workey`

On notera que cette propriété est également utilisée lors de la construction des liens d'accès aux documents exposés dans le résultat des recherches.

Evènements

Les propriétés suivantes sont XXXXXXXXXX.

Propriété	Description	Remarque
<code>com.clog.workey.events.preDeployProcess</code> <code>com.clog.workey.events.postDeployProcess</code> <code>com.clog.workey.events.preAttachment</code> <code>com.clog.workey.events.postAttachment</code> <code>com.clog.workey.events.preNotification</code> <code>com.clog.workey.events.preSubmitDocument</code> <code>com.clog.workey.events.postSubmitDocument</code> <code>com.clog.workey.events.preDeleteDocument</code> <code>com.clog.workey.events.postDeleteDocument</code>	cf. Les évènements pré-définis de Workey .	-> ces propriétés doivent être renseignées avec des nom de classe java pleinement qualifiés. -> Il est possible de spécifier une liste de noms de classes pour un même évènement, séparés par des ',' (virgule). Les classes d'évènement seront alors appelées de façon chaînée, dans l'ordre de la liste.

Fonctionnalités diverses

Les propriétés suivantes sont **optionnelles**.

Propriété	Description	Remarque
<code>com.clog.workey.admin.adminUrl</code> (à partir de Workey 6.4.0)	-> Permet de spécifier explicitement l'URL d'accès à la console d'administration depuis le serveur.	Dans le cadre du déploiement de Workey dans votre architecture réseau, vous pouvez être amené à modifier le contexte sous lequel est accessible la console d'administration. Certaines fonctionnalités du moteur s'appuient sur la console d'administration et doivent être en mesure d'y accéder. -Exemple : mécanisme de transformations XSL pour le remplacement d'acteur, le détail d'acteur, ...). -Pour assurer leur bon fonctionnement, il vous sera nécessaire d'ajouter cette propriété (ou de la décommenter si elle est déjà présente) et de renseigner sa valeur avec l'URL de la console d'administration accessible depuis le serveur Workey. -> Par défaut cette propriété est absente/commentée.

Propriété	Description	Remarque
com.clog.workey.archive.debug.docLimit	-> Permet de restreindre arbitrairement le nombre de document à archiver à chaque itération.	La limite du nombre de document à archiver doit être un entier non nul . Une valeur nulle ou négative ne spécifie aucune limite. -> La valeur par défaut : -1
com.clog.workey.index.analyzer	-> Permet de spécifier une classe d'analyse spécifique pour l'indexation Lucene .	Renseigner la propriété avec le nom de la classe Java (pleinement qualifié). -> La valeur par défaut : com.clog.workey.search.FrenchSnowballAnalyzer
com.clog.workey.engine.actorsService.newActorOnRequestRole (à partir de Workey 6.3.6)	-> Permet de débrayer la création automatique d'acteur , en cas de « Rôle par défaut », lorsqu'un nouvel utilisateur se connecte à l'application.	Pour désactiver la création automatique, renseigner explicitement la propriété avec no -> Dans le cas contraire, la fonctionnalité est toujours activée
com.clog.workey.engine.Delegations.taskTransfer	-> Permet d' autoriser le transfert des tâches dans le cadre des délégations.	Pour activer le transfert des tâches, renseigner la propriété avec true -> La valeur par défaut : false
com.clog.workey.engine.Delegations.allowsForwardedDelegations	-> Permet d' autoriser la propagation des délégations .	Pour interdire la propagation des délégations, renseigner la propriété avec false -> La valeur par défaut : true
com.clog.workey.engine.SubmitDoc.takeInChargeRecorder	-> Permet d' activer le suivi de la prise en charge des documents.	Pour que les prises en charge/libérations des documents soient tracées, renseigner la propriété avec true . -> La valeur par défaut : false
com.clog.workey.engine.QueryWorkflow.initialView	-> Permet de spécifier la vue Workey affichée par défaut .	La vue souhaitée doit être désignée par son designer name .-> Par défaut, la vue standard « A Faire » sera affichée.
com.clog.workey.engine.AttachmentRepository.fileExtension	-> Permet d'associer une extension aux fichiers des pièces jointes .	Dans le cas où une extension est spécifiée, le caractère '.' (point) ne doit pas y être présent.-> Par défaut, Workey n'utilise aucune extension pour les fichiers de pièce jointe. -> La valeur par défaut : false
com.clog.workey.engine.AttachmentRepository.purgeEnabled	-> cf. administration des pièces jointes .	
com.clog.workey.engine.AttachmentRepository.directoryStrategy	-> cf. stratégie de gestion des fichiers . Deux stratégies sont disponibles:StandardDirectoryStrategyHashDirectoryStrategy	-> La valeur par défaut : StandardDirectoryStrategy
com.clog.workey.engine.ThumbnailRepository.fileExtension	-> Permet d'associer une extension aux fichiers des vignettes .	Dans le cas où une extension est spécifiée, le caractère '.' (point) ne doit pas y être présent.-> Par défaut, Workey n'utilise aucune extension pour les fichiers des vignettes. -> La valeur par défaut : false
com.clog.workey.engine.ThumbnailRepository.purgeEnabled	-> cf. administration des vignettes .	
com.clog.workey.engine.ThumbnailRepository.directoryStrategy	-> cf. stratégie de gestion des fichiers . Deux stratégies sont disponibles:StandardDirectoryStrategyHashDirectoryStrategy	-> La valeur par défaut : StandardDirectoryStrategy
com.clog.workey.engine.ThumbnailRepository.scalingMethod	-> cf. génération des vignettes .	-> La valeur par défaut : AUTOMATIC
com.clog.workey.engine.ThumbnailRepository.MIMType	-> cf. type MIME des vignettes . Le type MIME des vignettes générées peut être l'un des suivants:image/png image/gif image/jpeg image/pjpeg	-> La valeur par défaut : image/png
com.clog.workey.engine.ThumbnailRepository.useAntiAliasing	-> Permet de désactiver l'usage de l' anti-crénelage pour les vignettes .	-> Renseigner cette propriété avec la valeur false pour désactiver l'anti-crénelage (anti-aliasing). -> La valeur par défaut : true
com.clog.workey.engine.defaultLanguage (à partir de Workey 6.0.2)	-> Permet de définir explicitement la langue par défaut pour le moteur Workey. -> cf. Multilinguisme dans l'interface Web standard .	Si la propriété n'est pas définie alors la langue sera celle définie par la variable Java : user.language -> Cette langue doit faire partie de la liste des langues supportées.
com.clog.workey.engine.supportedLanguages (à partir de Workey 6.0.2)	-> Permet de spécifier la liste des langues supportées par le moteur Workey.-> cf. Multilinguisme dans l'interface Web standard .	-> Cette liste doit contenir la langue par défaut utilisée par le moteur.
com.clog.workey.patch.ApplySpeedy.dropConflictingIndexes	-> Permet d' autoriser explicitement la suppression des index conflictuels . A chaque démarrage du serveur Workey, une vérification des index présents sur les tables de la base socle est effectuée afin de s'assurer que l'ensemble des index requis sont présents et que leur définition est bien celle attendue.	Dans les cas où:un index de même nom existe déjà mais a une définition différente (colonne(s) et/ou ordre des colonnes différent),un index de nom différent existe déjà et correspond à la définition requise,-> alors, cet index <i>conflictuel</i> doit être supprimé pour permettre de créer celui attendu.A défaut de pouvoir supprimer les index <i>conflictuels</i> , les index requis ne pourront pas être mis en place. -> Renseigner cette propriété avec la valeur true pour autoriser la suppression . -> La valeur par défaut : false
com.clog.workey.web.CustomSkinUrl	-> Permet de spécifier l' URL d'une servlet desservant des fichiers personnalisés de CSS et/ou JS utilisables par l'interface Web standard.	Propriété renseignée avec le contexte d'une <i>servlet</i> déployée séparément (fichier .war distinct) dans le serveur <i>Apache Tomcat</i> .
com.clog.workey.web.main.connector.documentClose	-> Permet de spécifier une redirection suite à la soumission d'un document Workey dans l'interface Web standard. -> Cette propriété relève de développements spécifiques.	Propriété renseignée avec un nom de classe Java (pleinement qualifié).