Version 8.3.10397

Gargantua 8 nouveautés

Ce document est la propriété de SIATEL.
Il ne peut être reproduit ou communiqué sans l'autorisation écrite de SIATEL.

Table des matières

Introduction 7

Interface utilisateur 8

Généralités 8

Connexion à l'aide d'une double authentification 8

Nom d'agrafe basé par défaut sur le premier document agrafé 9

Affichage du texte reconnu par l'OCR suite à un encadrement de zone 9

Déblocage des documents ouverts dans OnlyOffice 11

Opérations interdites sur les documents signés 11

Contrôle de format sur un champ de type téléphone 11

Commande de bascule en plein écran 12

Indication du nombre de filtres appliqués 12

Signature et certification de documents 13

Compatibilité ALT 13

Fonctions de zoom sur un document affiché en mode Signature 13

Création de nouvelles signatures 14

Affichage d'une zone « Signez ici » en l'absence de signature prédéfinie 15

Prévisualisation des signatures prédéfinies et ouverture de la boîte de dialogue de création d'une nouvelle signature 16

Commande « Afficher/Masquer les outils de signature » 16

Module Portails 17

Choix du portail à ouvrir à partir d'un menu déroulant 17

Améliorations des performances d'affichage des documents favoris et des derniers documents consultés 17

Module Bases documentaires 18

Commande « Diffusion » et options de diffusion disponibles 18

Tri des dossiers 23

Personnalisation de l'ordre des favoris par glisser-déposer 25

Information sur la taille d'un dossier 26

Module Recherche multicritères 26

Opérateurs à disposition pour les critères de type date et date/heure 26

Opérateurs à disposition pour les critères Propriétaire, Modifié par, Requis par, Signé par, Validé par 28

Opérateurs à disposition pour les critères alphanumériques 28

Options à disposition pour les modèles de recherche sauvegardés 28

Champs d'indexation à afficher dans la liste des résultats 30

Module Tâches 30

Panneau d'affichage des documents 30
Liste des tâches 31
Aide en ligne relative à un processus 32
Panneau de recherche multicritères 32
Module Acquisition de documents 35
Etape Capture 35
Modernisation des paramètres de découpe 38
Etape Découpe 39
Etape Indexation 40
Aide en ligne 41
Module Carnet d'adresses 41
Module Outils 41
Gestion des documents bloqués 41
Gestion des liens de téléchargement 42
Module Supervision des processus 43
Filtres de recherche disponibles 43
Module Paramétrage 43
Rubrique Personnalisation des menus 43
Module Gestionnaire 44
Modifier le propriétaire 44
Suggérer un mot de passe fort aléatoire 44

CONSOLE D'ADMINISTRATION 45
Paramètres du serveur 45
Plugins 45
Gestion des utilisateurs 47
Privilège « Double authentification » 47
Gestion des rôles 48
Affectation en un clic de tous les utilisateurs à un rôle 48
Personnalisation de l'interface par profil 49
Personnalisation des menus 49
Profils personnels 50
Gestion des bases documentaires 51
Onglet Extensions 51
Gestion des processus 61
Gestion des modèles de document 65
Gestion des portails 65
Rattachement d'un portail à une thématique 65
Personnalisation des noms des portails 65
Configuration des modules 66
Module Tâches 66

Module Portails 70
Modèles de notification 71
Envoi du code de sécurité pour l'authentification à deux facteurs 71
Tâches IntelliScan 72
Compatibilité avec le driver du scanner Canon DR-G2090 72
APIs scripting serveur 72
Modification du champ d'indexation date/heure 72
Amélioration de l'énumération des rôles et des groupes d'un utilisateur 73
Méthode Ged.deleteFolder 73
Intégration d'Apache OpenNLP comme moteur de détection de la langue 73
Carnet d'adresses v2 73
Support pour UUID pour les commandes GedLibrary.READ_FOLDER/DOCUMENT 91

MODULES COMPLÉMENTAIRES ET AUTRES OUTILS 94
N° de version des modules complémentaires 94
N° des versions des autres outils 94
Module complémentaire commun à tous les navigateurs 95
Interface d'ajout de documents : taille des icônes fonction de la résolution d'écran 95
OCR 96
Fichiers compressés 99
Inclusion de l'installation de l'OCR Tesseract dans l'installation du module complémentaire commun aux navigateurs 100
Actualisation de la liste des scanners disponibles 100
Optimisations 101
Fichiers PDF 102
Compatibilité avec le driver du scanner Canon DR-G2090 103
IntelliScan 105
Compatibilité avec le driver du scanner Canon DR-G2090 105
EasyScan 105
Compatibilité avec le driver du scanner Canon DR-G2090 105
CustomScan 105
Compatibilité avec le driver du scanner Canon DR-G2090 105
Modules complémentaires bureautiques 105
Fonction d'affichage de document disponible depuis un formulaire 105

EXTENSIONS/MODULES SERVEUR 107
Module Moteur de classement 107
Configuration d'une temporisation 107
Module Gestionnaire 107
Changement du propriétaire 107

Génération d'un mot de passe fort aléatoire 108

AUTRES DÉVELOPPEMENTS 109

Serveur OCR 109

Détection automatique de la langue d'un document 109

Nettoyage des fichiers orphelins après un temps prédéterminé 118

Prise en compte de l'ordre des zones 119

Setup d'installation de l'OCR Tesseract 119

Mise à jour du setup de l'OCR Tesseract à la version 5.3 119

Installation optionnelle d'autres langues que celles par défaut 119

Mise à jour de la librairie PDFBox 120

Introduction

Ce document détaille les fonctionnalités amenées par la version 8.3 de Gargantua.

Le premier chapitre, dédié à l'**Interface utilisateur,** décrit les outils désormais disponibles pour les utilisateurs finaux de Gargantua.

Les autres chapitres sont spécifiquement conçus pour les administrateurs de la solution, fournissant les informations essentielles de gestion et de configuration.

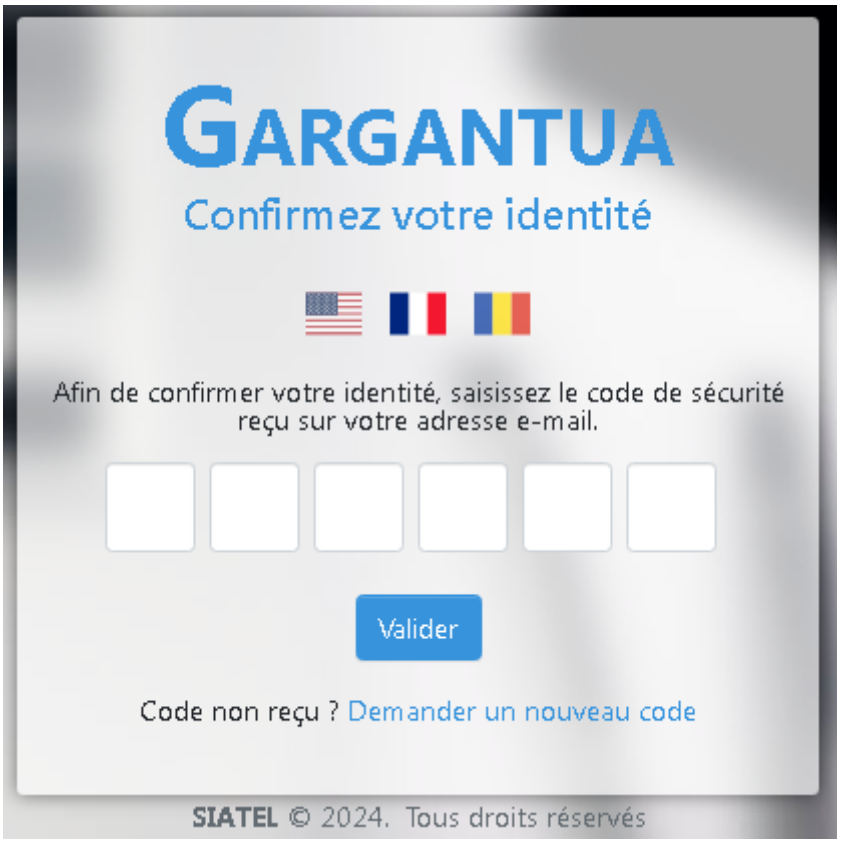
L'équipe de Siatel reste à votre entière disposition pour toute clarification ou complément d'information, et vous souhaite une bonne lecture !

Interface utilisateur

Généralités

Connexion à l'aide d'une double authentification

Pour les utilisateurs ayant la propriété **Double authentification**, la connexion à Gargantua nécessite une authentification à deux facteurs (ou double authentification) : après avoir saisi leur mot de passe, les utilisateurs doivent renseigner un code à usage unique et à durée limitée. Ils reçoivent ce code par SMS ou par mail, selon que leur numéro de mobile est renseigné ou non au niveau de leur compte utilisateur Gargantua. En l'absence de ces deux informations, la double authentification ne peut pas fonctionner.



Nom d'agrafe basé par défaut sur le premier document agrafé

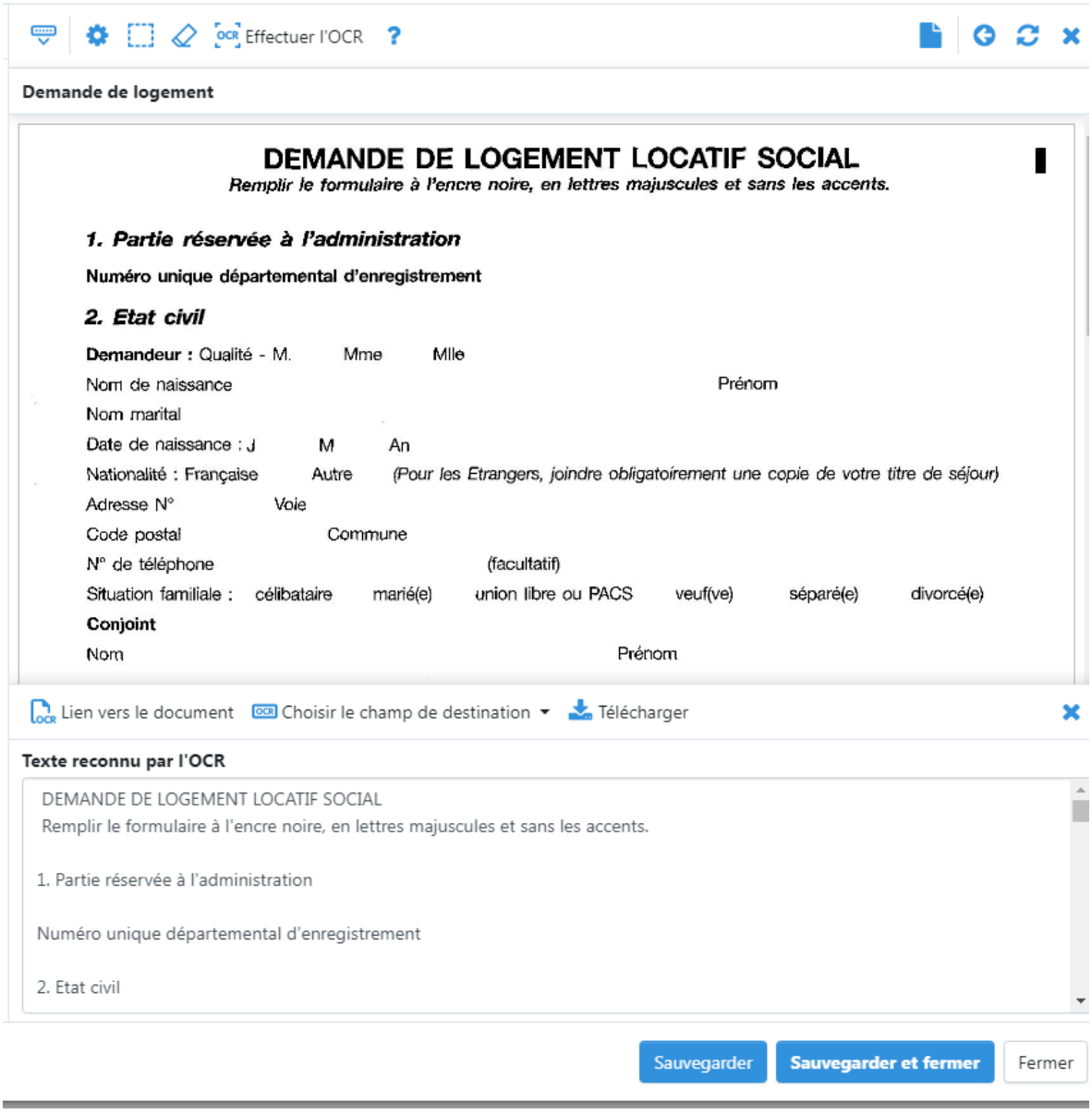
A la création d'une agrafe, par défaut, son nom est maintenant initialisé avec le nom du premier document qu'elle contient.

> *Quand l'interface d'un profil d'utilisateur est personnalisée avec l'option* ***Préremplir le formulaire du document avec les champs d'indexation du dossier parent****, la procédure adoptée est la suivante :*
> - *Les noms des documents inclus dans l'agrafe sont inchangés,*
> - *Les valeurs des autres champs d'indexation des documents sont actualisées en fonction de celles définies au niveau du dossier parent.*

Affichage du texte reconnu par l'OCR suite à un encadrement de zone

Désormais, le texte reconnu est affiché dans un panneau sous le document.

Lorsqu'un utilisateur encadre une zone dans un document pour activer l'OCR, le texte identifié par l'OCR s'affiche sous le document. Cela permet à l'utilisateur de garder le document visible pour vérifier la précision du texte reconnu et d'y apporter des corrections, si nécessaire.



Les boutons **Lien vers le document** et **Choisir le champ de destination** s'affichent ou non en fonction du contexte dans lequel est réalisé l'OCR. Les fonctions OCR sont en effet disponibles dans de nombreuses fonctions (en GED, dans des processus, dans des formulaires...).

> ***Le panneau d'affichage du résultat OCR a une taille fixe. Il n'est pas possible de l'avoir plus grand car dans le cas d'écrans de petite taille, comme sur une tablette par exemple, on risquerait de ne plus voir du tout le document. En plus, il existe des cas où la visionneuse est affichée dans un mode modal ce qui induit également que sa hauteur peut être inférieure à celle de la totalité de l'écran.***

Déblocage des documents ouverts dans OnlyOffice

Lorsque des documents sont ouverts en simultané par plusieurs utilisateurs à l'aide d'OnlyOffice, il n'est pas possible de définir par qui ils sont bloqués. Dans ce cas, Gargantua indique qu'ils sont bloqués par Système. Cela a pour conséquence que seuls les membres du groupe DBA peuvent débloquer ces documents.

Le déblocage s'effectue avec la commande **Débloquer**.

Opérations interdites sur les documents signés

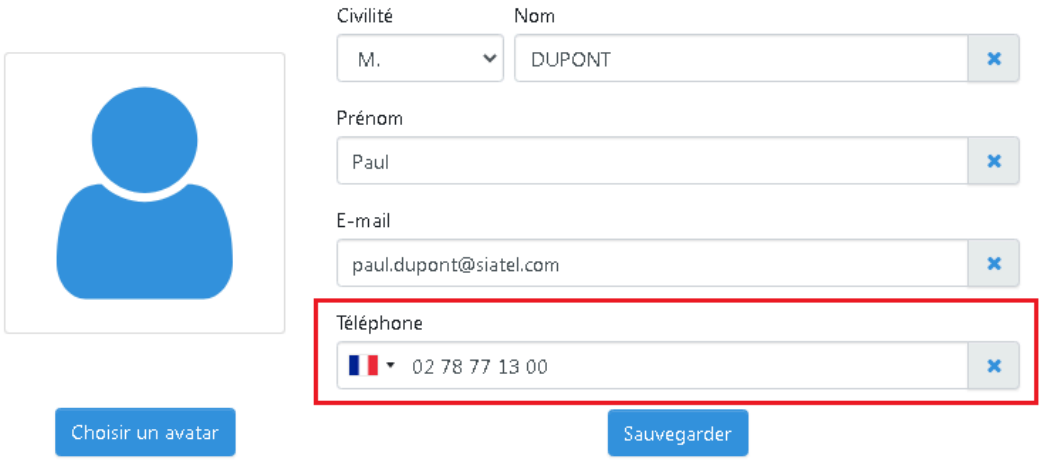
Certaines opérations ou commandes sont interdites sur les documents signés :

- **Remplacer ou créer une version,**
- **Restaurer** une version précédente grâce à la boîte de dialogue **Gérer les versions** (accessible par le menu contextuel affiché suite à un clic droit sur un document) n'est pas possible.
- Restaurer une version précédente lors de la **Suppression** d'un document n'est pas possible. Dans ce cas, le document est directement supprimé et la boîte de dialogue affichant le choix entre la suppression du document et la restauration d'une ancienne version ne s'affiche pas.
- **Démarrer un circuit de validation**
- **Ouvrir le document dans son application d'origine**. Le document ne peut être qu'ouvert en lecture seule. Ce mode de fonctionnement est soumis à l'installation des derniers modules complémentaires.
- Depuis la corbeille, la **Restauration** d'une version supprimée vers son emplacement original est impossible si le document original a été signé entre temps.

Contrôle de format sur un champ de type téléphone

Depuis la version 8.3, le contrôle de format d'un n° de téléphone lors de la saisie est étendu aux modules et composants suivants :

- **Paramétrage** > **Modifier vos données personnelles**



- Circuits de signature
- Module **Gestionnaire**

Commande de bascule en plein écran

Sur certaines interfaces, il est possible de passer ou de quitter l'affichage en plein écran. On dispose pour cela de deux boutons :

- Basculer en plein écran
- Quitter le plein écran

Indication du nombre de filtres appliqués

Pour tous les affichages Gargantua qu'il est possible de filtrer, le nombre de filtres appliqués est indiqué grâce à une pastille disposée sur le bouton de sélection des filtres

.

Signature et certification de documents

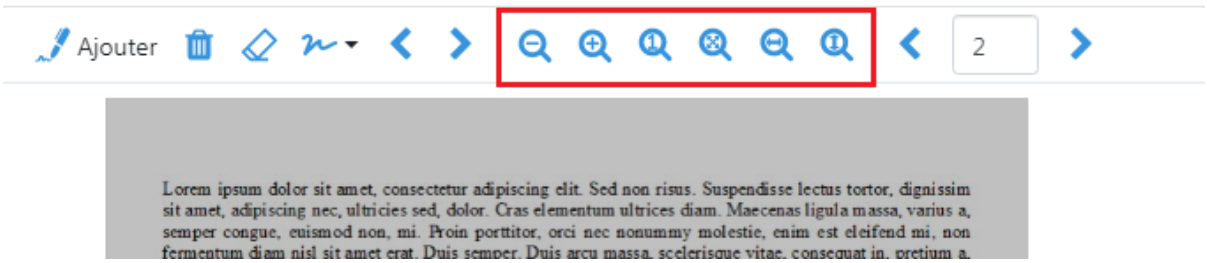
Compatibilité ALT

Pour renforcer la validité des certificats de signature, la version 8.3 amène des améliorations qui garantissent la compatibilité avec le format ALT. Cette mise à jour concerne les signatures et certifications utilisant des certificats codés en LT, plutôt qu'en DER.

Fonctions de zoom sur un document affiché en mode Signature

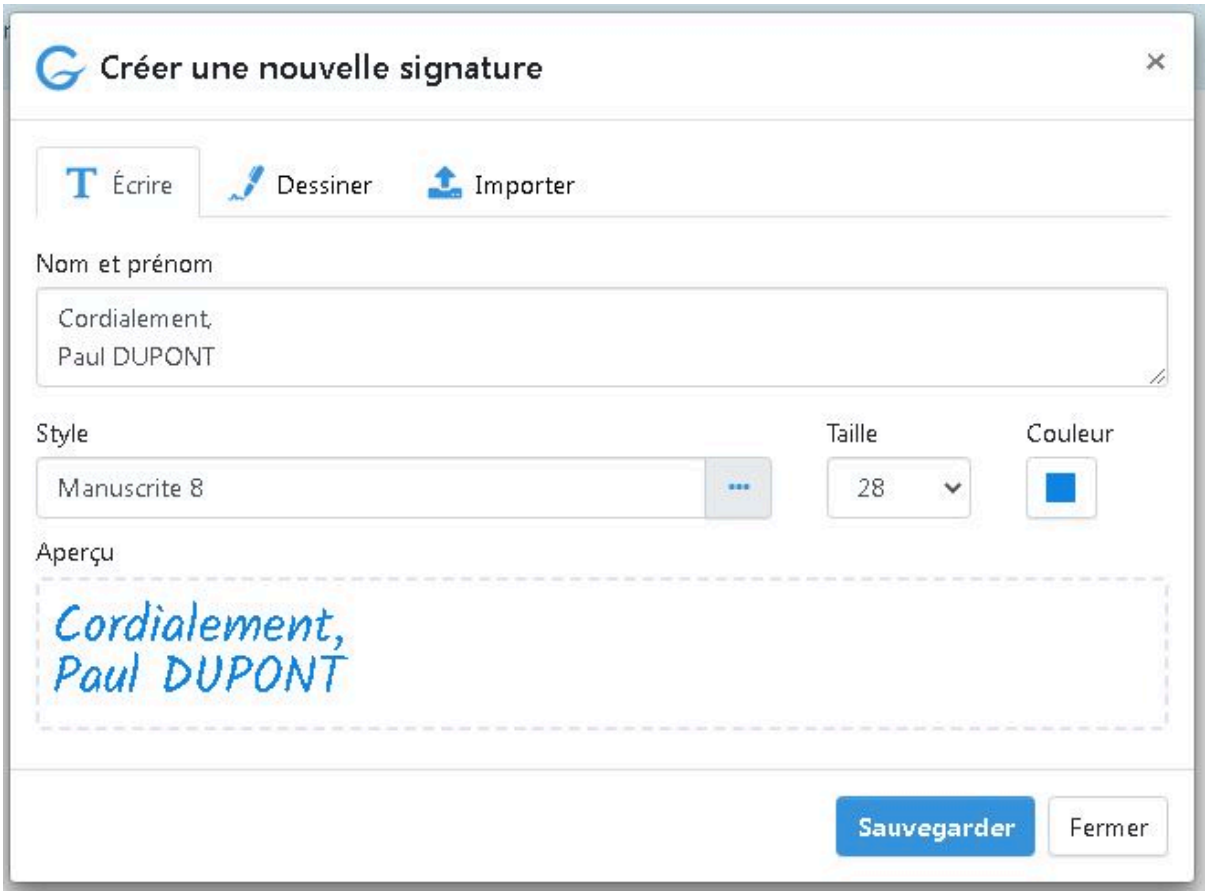
Les commandes de zoom, introduites lorsqu'un document est affiché dans l'interface en mode Signature, permettent aux utilisateurs d'agrandir ou de réduire la vue du document pour une meilleure visibilité et précision lors de l'apposition de leur signature.

Cette fonctionnalité, similaire à celle historiquement présente dans les visionneuses d'images et d'OCR, est disponible dans les processus, dans les circuits de signature, dans le module e-parapheur, ainsi que pour les commandes de signature. Elle facilite l'examen détaillé des documents avant l'apposition d'une signature ou d'un paraphe.



Création de nouvelles signatures

La boîte de dialogue de création de signature propose trois possibilités : la saisir sous forme de tapuscrit, la dessiner, ou l'importer.



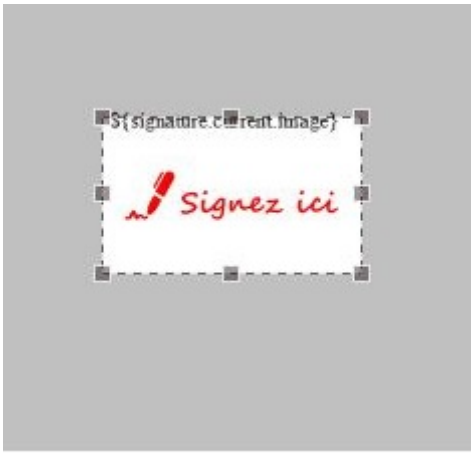
Cette boîte de dialogue est utilisée dans l'aperçu de signature, permettant ainsi à l'utilisateur de créer une nouvelle signature directement sur le même écran sans avoir à naviguer vers la page **Paramétrage > Gérer vos signatures.** Cette boîte de dialogue est également disponible dans la page **Paramétrage > Gérer vos signatures** et dans le module **Parapheur** pour créer et modifier une signature existante.

Remarque : lors de l'utilisation de la commande **Ajouter une signature**, cette boîte de dialogue s'ouvre automatiquement quand l'utilisateur ne dispose d'aucune signature prédéfinie.

Nota : la commande **Ajouter une signature** peut ne pas apparaître, par exemple dans le cas d'un document envoyé dans un circuit de signature car le nombre de marqueurs peut avoir été prédéfini, interdisant leur ajout par l'utilisateur lui-même.

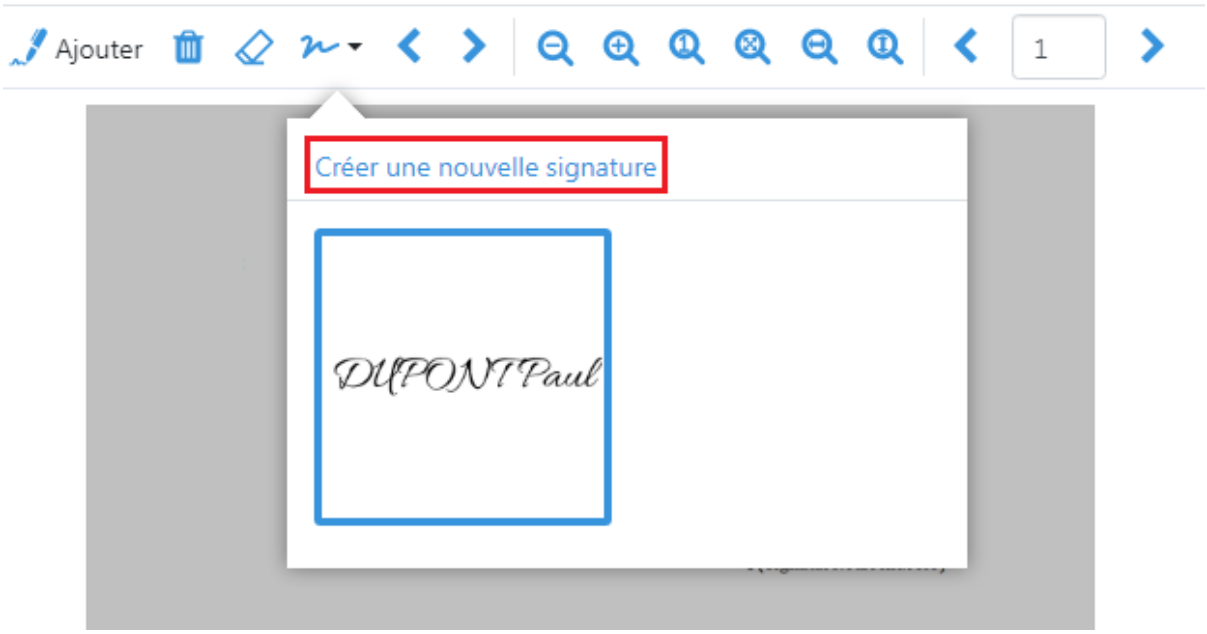
Affichage d'une zone « Signez ici » en l'absence de signature prédéfinie

Lors de l'ouverture d'un document à signer, si aucune signature n'a été préalablement définie par l'utilisateur, une zone intitulée « Signez ici » apparaît là où la signature est requise. En cliquant sur cette zone, l'utilisateur peut accéder à la boîte de dialogue pour créer une nouvelle signature



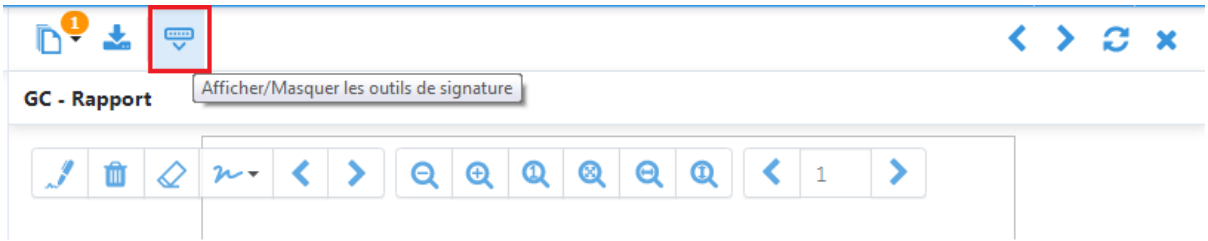
Prévisualisation des signatures prédéfinies et ouverture de la boîte de dialogue de création d'une nouvelle signature

Dès qu'un utilisateur dispose d'au moins une signature, une fenêtre permettant la prévisualisation des signatures existantes s'affiche, offrant également la possibilité d'accéder à la boîte de dialogue pour créer une nouvelle signature.



Commande « Afficher/Masquer les outils de signature »

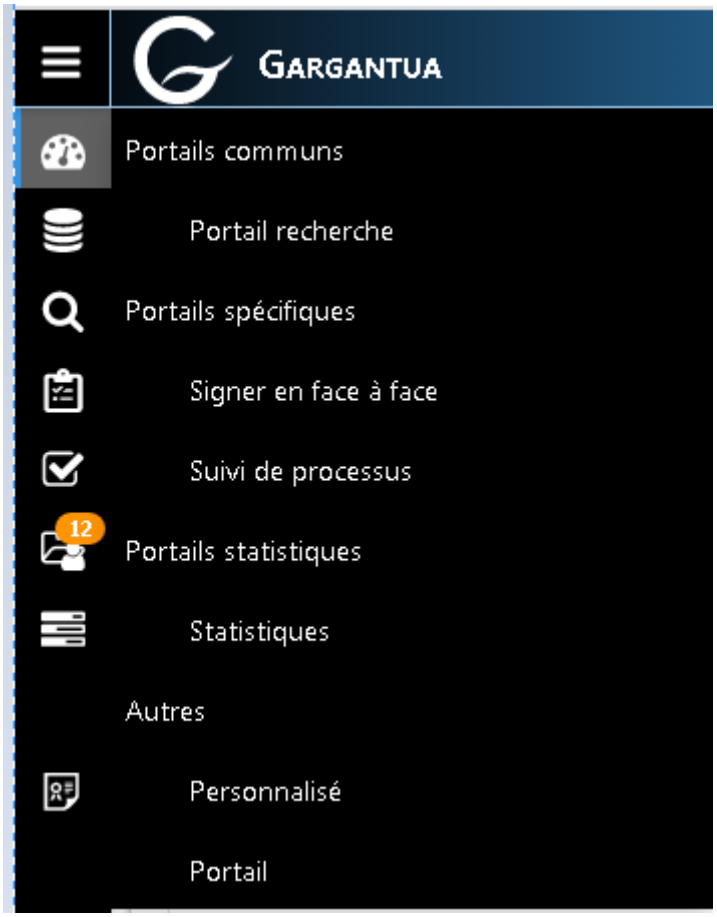
Un bouton **Afficher/Masquer les outils de signature** est présent sur la barre de commandes du panneau d'affichage des documents en mode Signature. Ce bouton permet d'afficher ou de masquer la barre contenant tous les outils de signature, au-dessus du document à signer.



Module Portails

Choix du portail à ouvrir à partir d'un menu déroulant

Lorsqu'un utilisateur possède au moins un portail associé à une thématique, un menu déroulant apparaît pour en faciliter l'accès, organisant les portails selon leur thématique respective. Les portails qui n'ont pas été assignés à une thématique spécifique par l'administrateur sont automatiquement inclus dans une catégorie générique dénommée **Autres**.



Améliorations des performances d'affichage des documents favoris et des derniers documents consultés

Les améliorations apportées dans les listes de documents favoris et derniers documents consultés conduisent à un affichage beaucoup plus rapide de la liste des ressources documentaires.

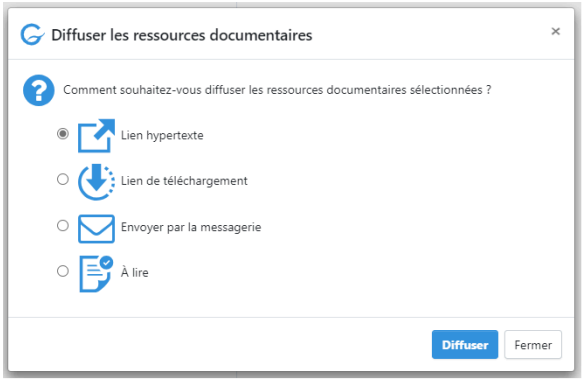
Module Bases documentaires

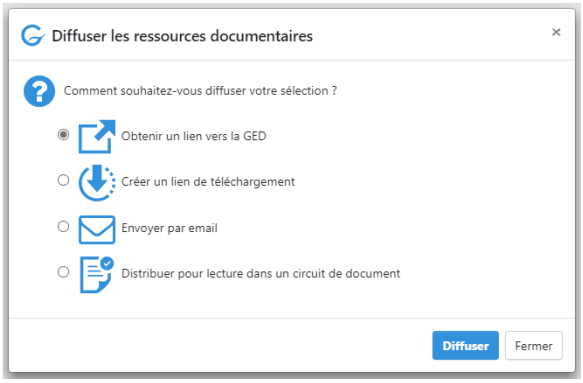
Commande « Diffusion » et options de diffusion disponibles

Avec la version 8.3, les différentes options de la commande **Diffuser** ont été modernisées et fluidifiées ; le vocabulaire a été adapté pour améliorer la clarté et l'efficacité. Auparavant connues sous les termes « Lien hypertexte », « Lien de téléchargement », « Envoyer par la messagerie », et « À lire », les options sont désormais respectivement intitulées :

- **Obtenir un lien vers le document en GED**
- **Obtenir un lien de téléchargement**
- **Envoyer par email**
- **Distribuer pour lecture dans un circuit de document**

Voir l'interface actuelle à droite, et l'ancienne interface à gauche.

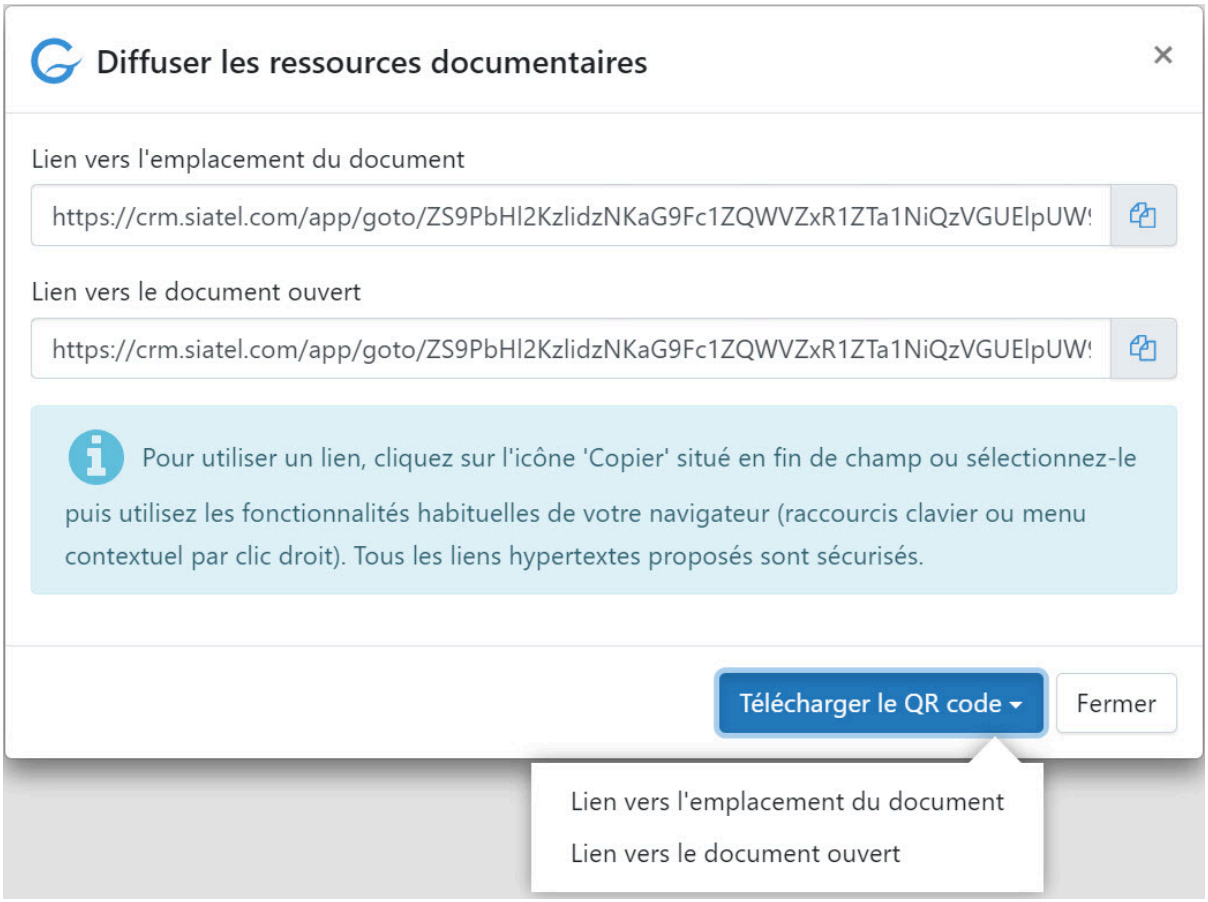




Diffusion d'un dossier ou d'un document via un lien vers la GED

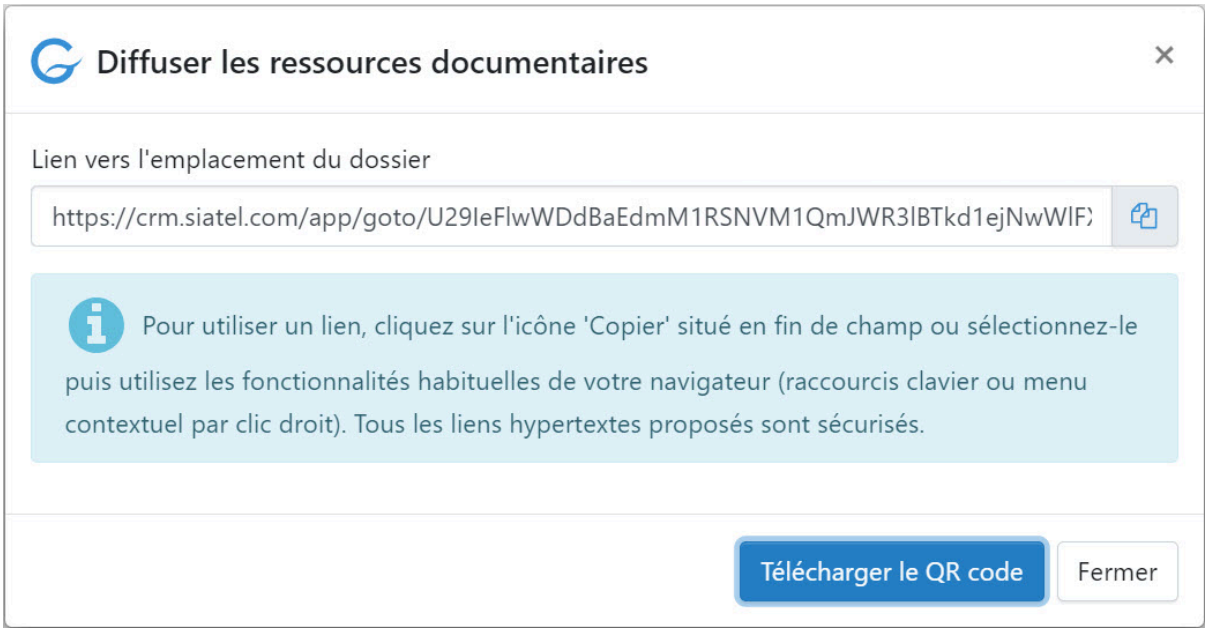
Depuis la version 8.3, la fenêtre de diffusion d'un document via un lien vers la GED permet d'obtenir :

- Soit un **lien vers l'emplacement de ce document**
- Soit un **lien vers le document ouvert**

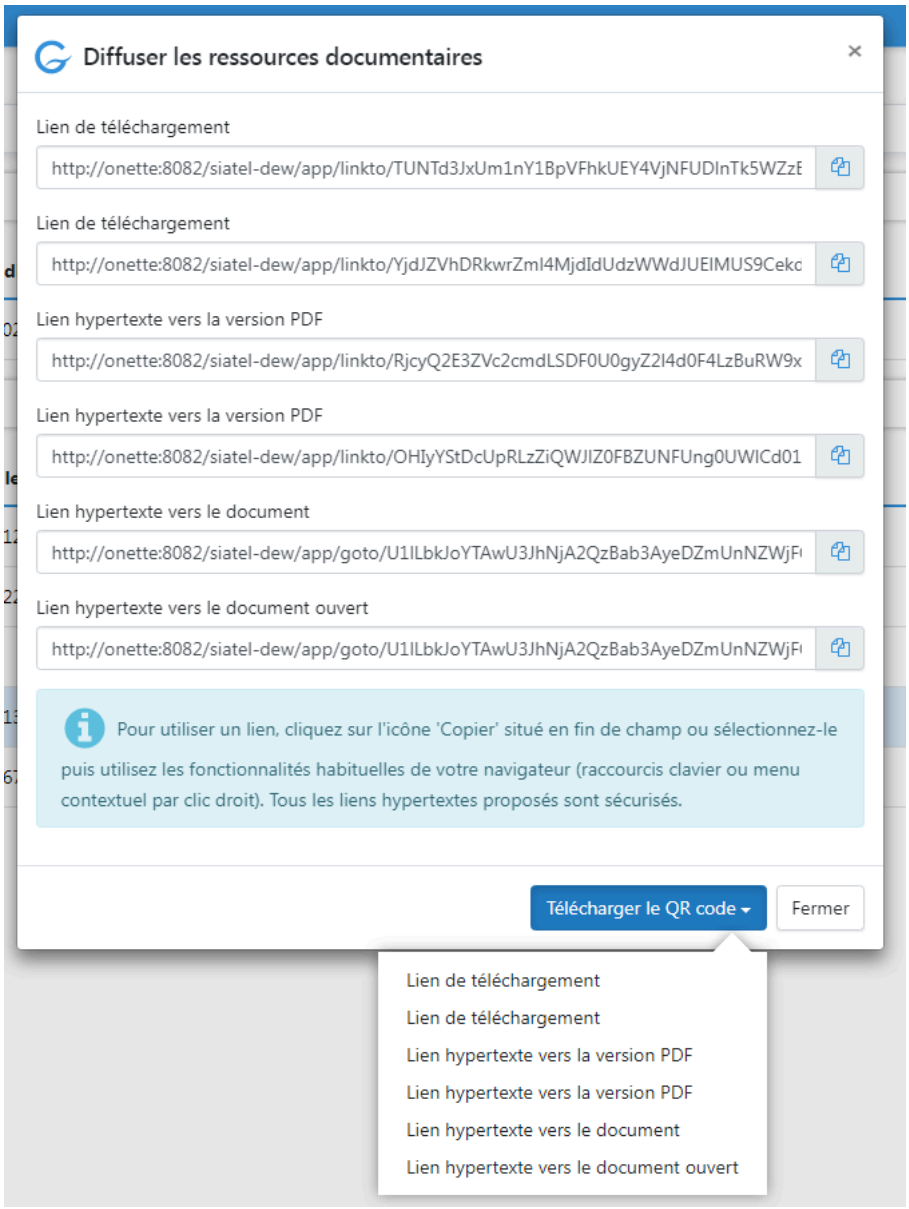


L'utilisateur peut alors copier ce lien pour le communiquer par mail par exemple à un autre utilisateur de la GED Gargantua.

En ce qui concerne la fenêtre de diffusion d'un dossier via un lien vers la GED, il est possible d'obtenir un lien vers l'emplacement du dossier, comme illustré ci-après :



Pour rappel, l'ancienne interface était comme suit :



Diffusion de documents via un lien de téléchargement et options disponibles

Lorsque l'option retenue est la diffusion de documents via un lien de téléchargement, plusieurs options peuvent être précisées :

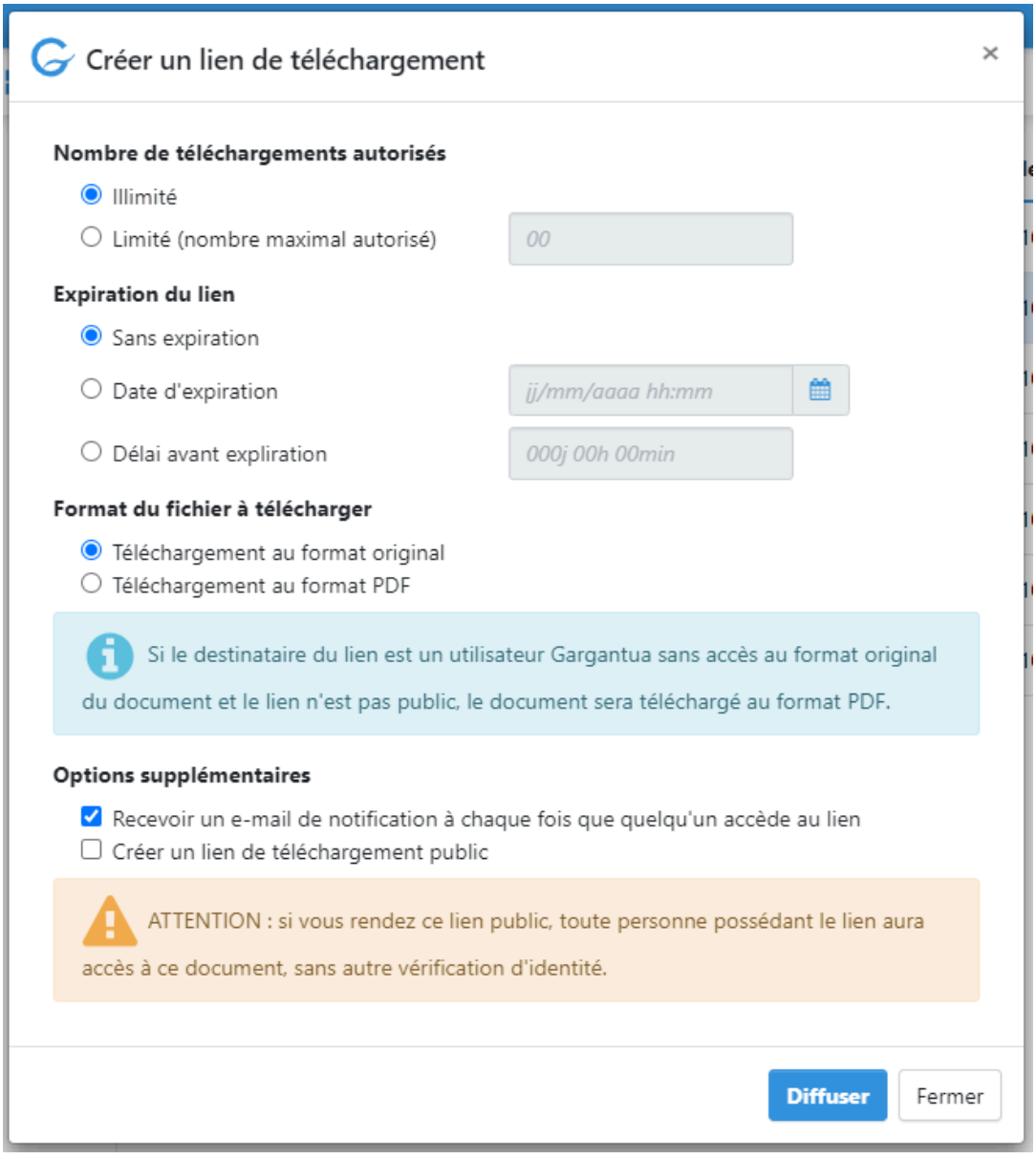
- **Le nombre maximum de téléchargements autorisé** (illimité par défaut)
- **La date ou le délai d'expiration du lien** (sans expiration par défaut)
- **Le format du fichier à télécharger** (format original par défaut)

Deux options supplémentaires permettent :

- De **recevoir une notification par mail** à chaque téléchargement
- De **créer un lien de téléchargement public** (utile pour partager le document avec des personnes n'ayant pas de compte utilisateur)

Remarque :

Si l'option **Créer un lien de téléchargement public** n'apparait pas dans votre interface, c'est que votre compte ne possède pas les privilèges nécessaires pour cela.



Notification avant le téléchargement d'un fichier via un lien externe

Dans un contexte de base documentaire sécurisée, lorsqu'un utilisateur clique sur un lien externe pour télécharger un fichier, il est d'abord redirigé vers une page intermédiaire. Cette page affiche un message lui demandant de patienter et indique que le téléchargement du fichier va bientôt commencer.

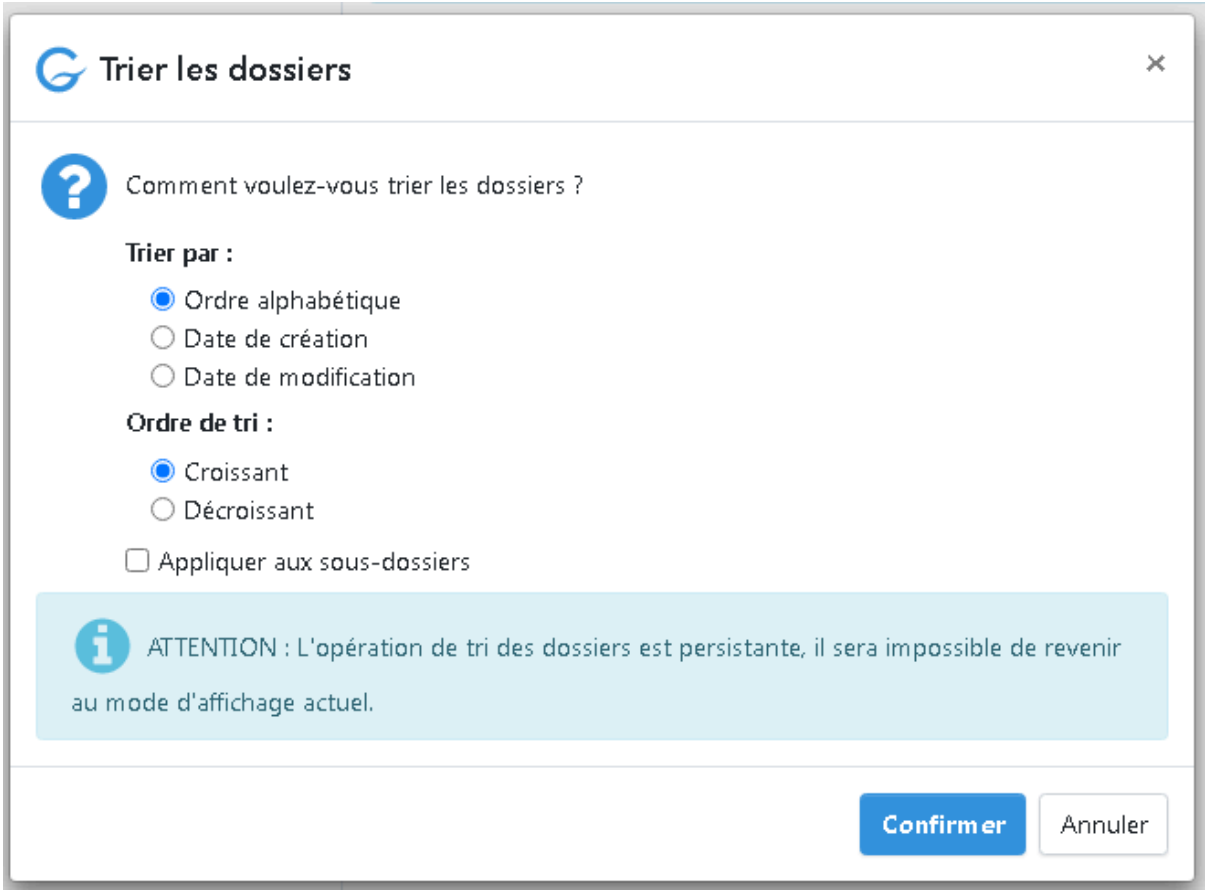
Tri des dossiers

Tri des dossiers dans la base active

Trois options de tri sont disponibles :

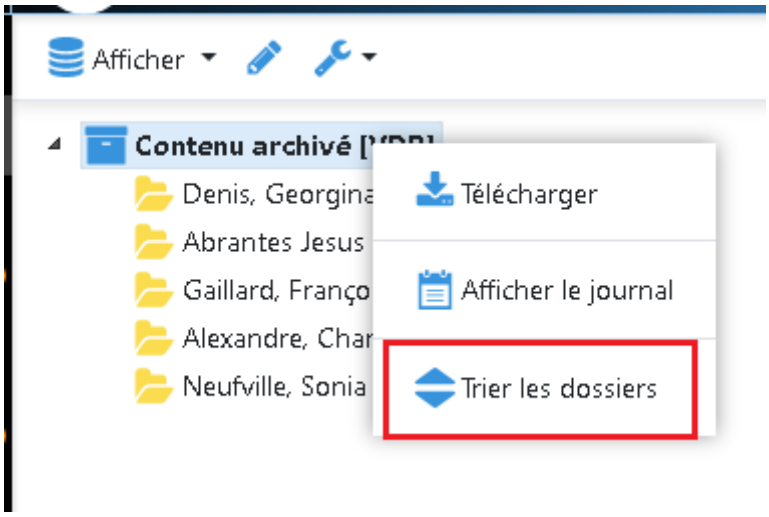
- Nom
- Date de création
- Date de modification des dossiers

Pour rappel, jusqu'en 8.3, il était seulement possible d'effectuer un tri par nom des dossiers.

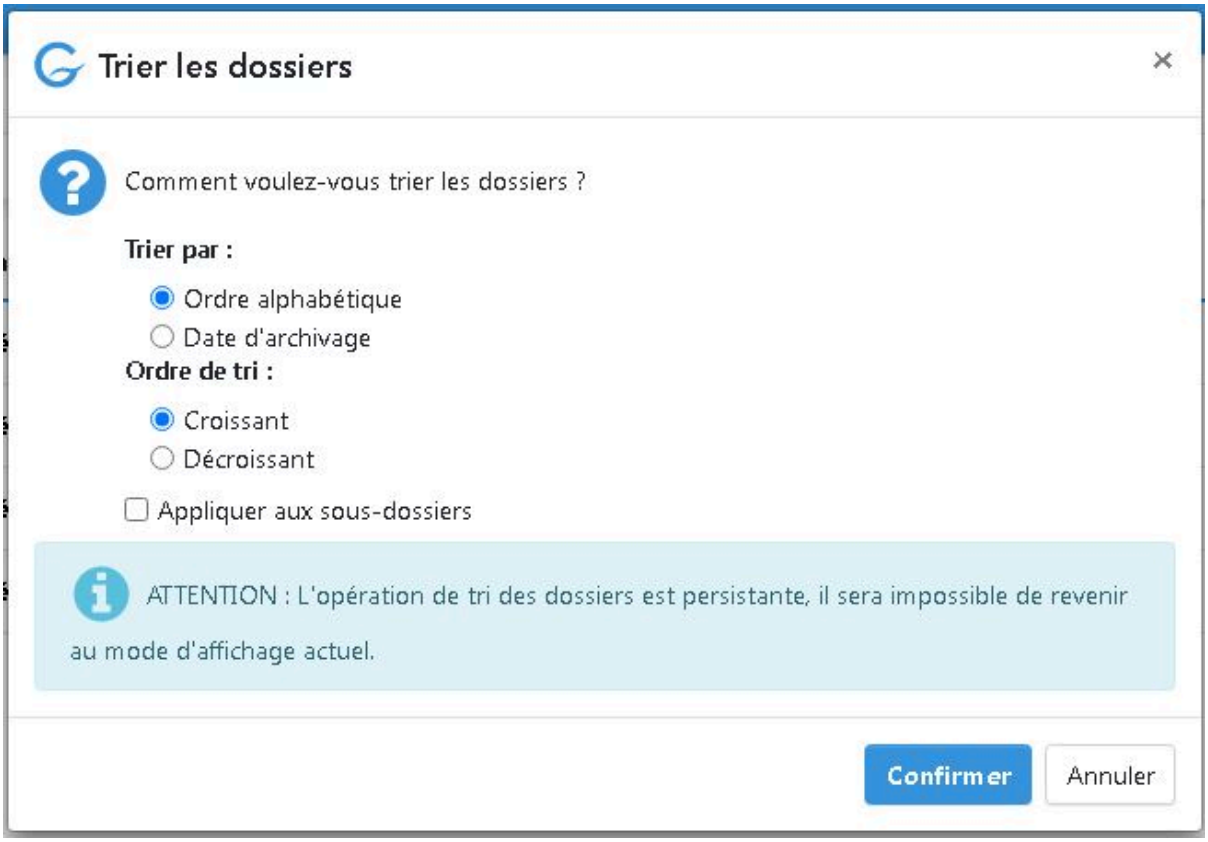


Tri des dossiers dans la base archive

La commande **Trier les dossiers**, conçue sur le modèle utilisé dans la base active, est accessible tant à la racine de la base archive qu'au sein de son arborescence.



Cette commande permet d'organiser les dossiers par ordre alphabétique de nom ou par ordre chronologique de la date d'archivage. Une option spécifique offre la possibilité d'étendre ce tri aux sous-dossiers.



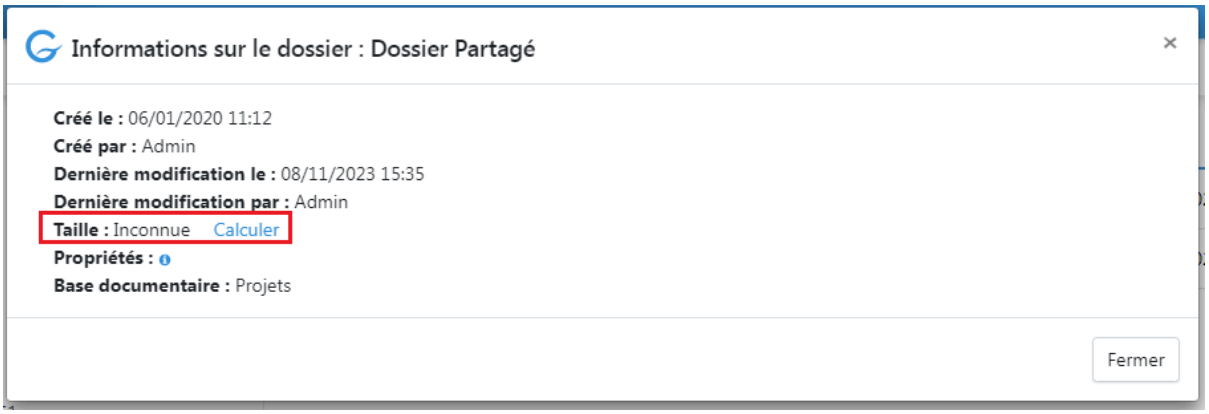
Personnalisation de l'ordre des favoris par glisser-déposer

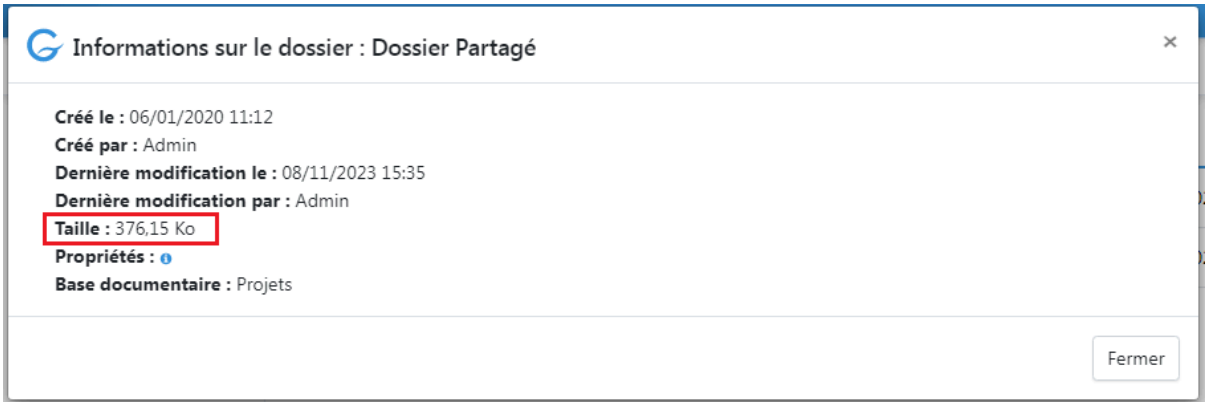
Les utilisateurs peuvent personnaliser l'ordre des documents dans leur dossier **Favoris** en utilisant la fonctionnalité de glisser-déposer. Cette action permet de repositionner facilement les documents à l'emplacement souhaité au sein de la liste des favoris.

> ***Ceci est disponible uniquement pour le dossier « Favoris ». Pour les dossiers « Derniers documents consultés » et « Documents verrouillés », le repositionnement est toujours désactivé.***

Information sur la taille d'un dossier

Dans la fenêtre **Informations sur le dossier**, la commande **Calculer** située à côté de l'indication **Taille** permet de connaître l'espace total occupé par le dossier et son contenu (documents et sous-dossiers) sur la zone de stockage. Notez que cette information est volatile et n'est pas mémorisée par le système.



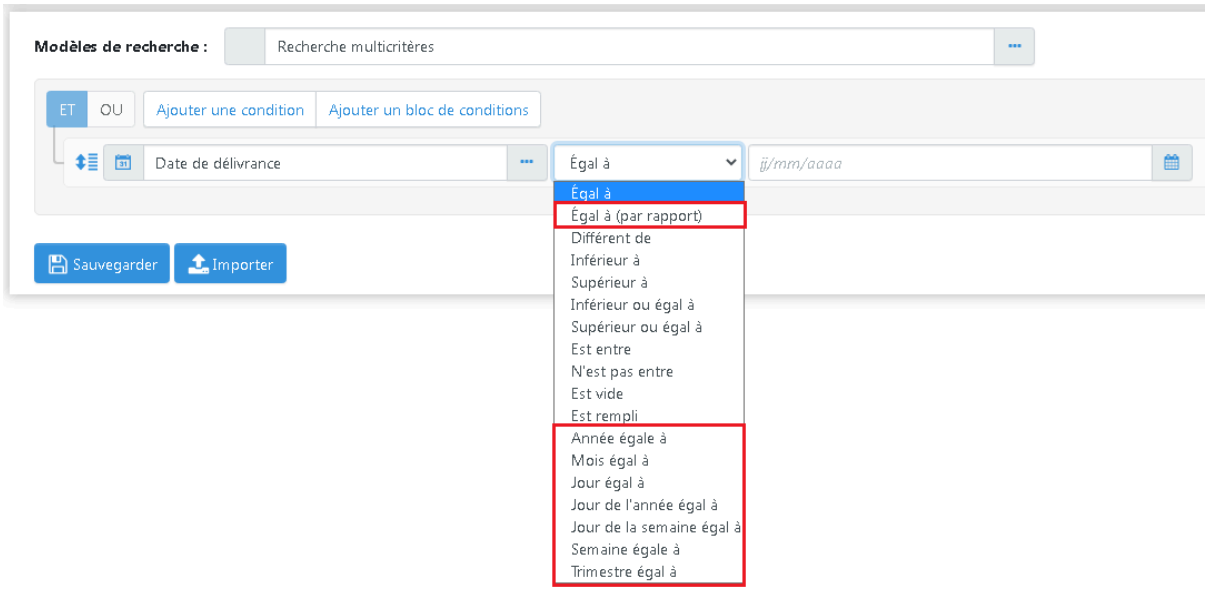


Module Recherche multicritères

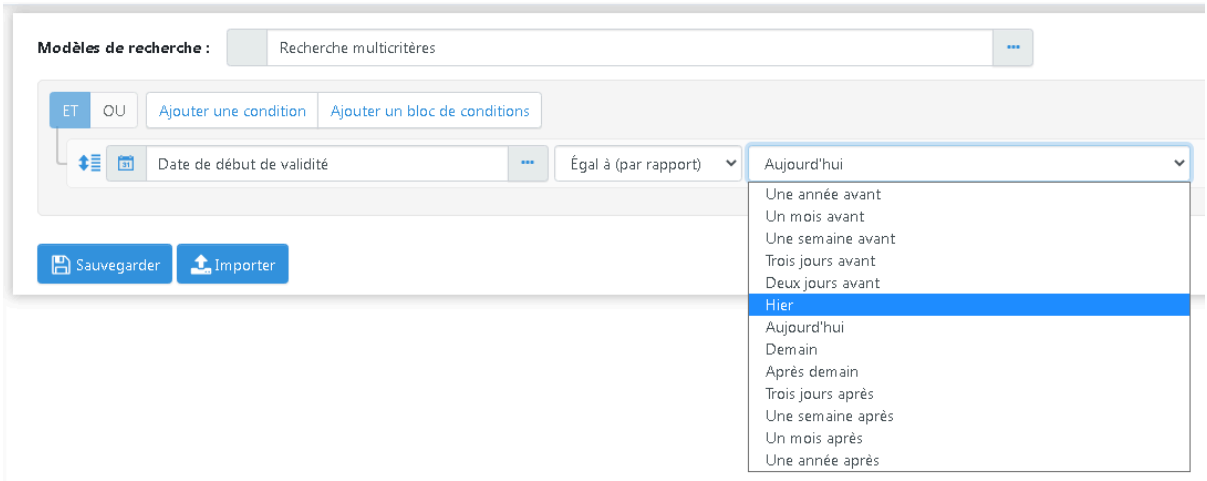
Opérateurs à disposition pour les critères de type date et date/heure

Certains opérateurs relatifs sont disponibles pour plusieurs types de champs d'indexation.

La version 8.3 élargit le choix des opérateurs pour les critères de type date et date/heure, comme illustré ci-après :

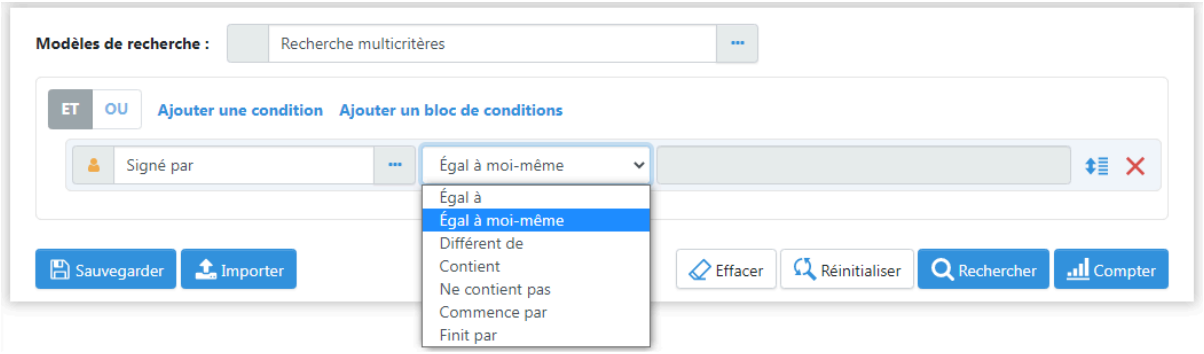


Pour l'opérateur **Égal à (par rapport)**, une liste des valeurs est proposée :



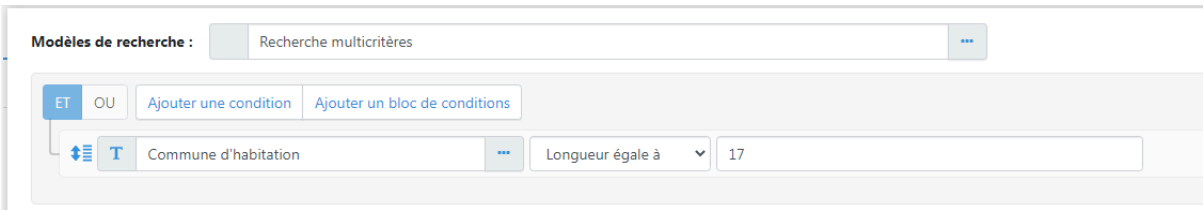
Opérateurs à disposition pour les critères Propriétaire, Modifié par, Requis par, Signé par, Validé par

La version 8.3 enrichit les options d'opérateurs pour les critères tels que **Propriétaire**, **Modifié par**, **Requis par**, **Signé par**, et **Validé par**, en introduisant l'opérateur **Égal à moi-même**.



Opérateurs à disposition pour les critères alphanumériques

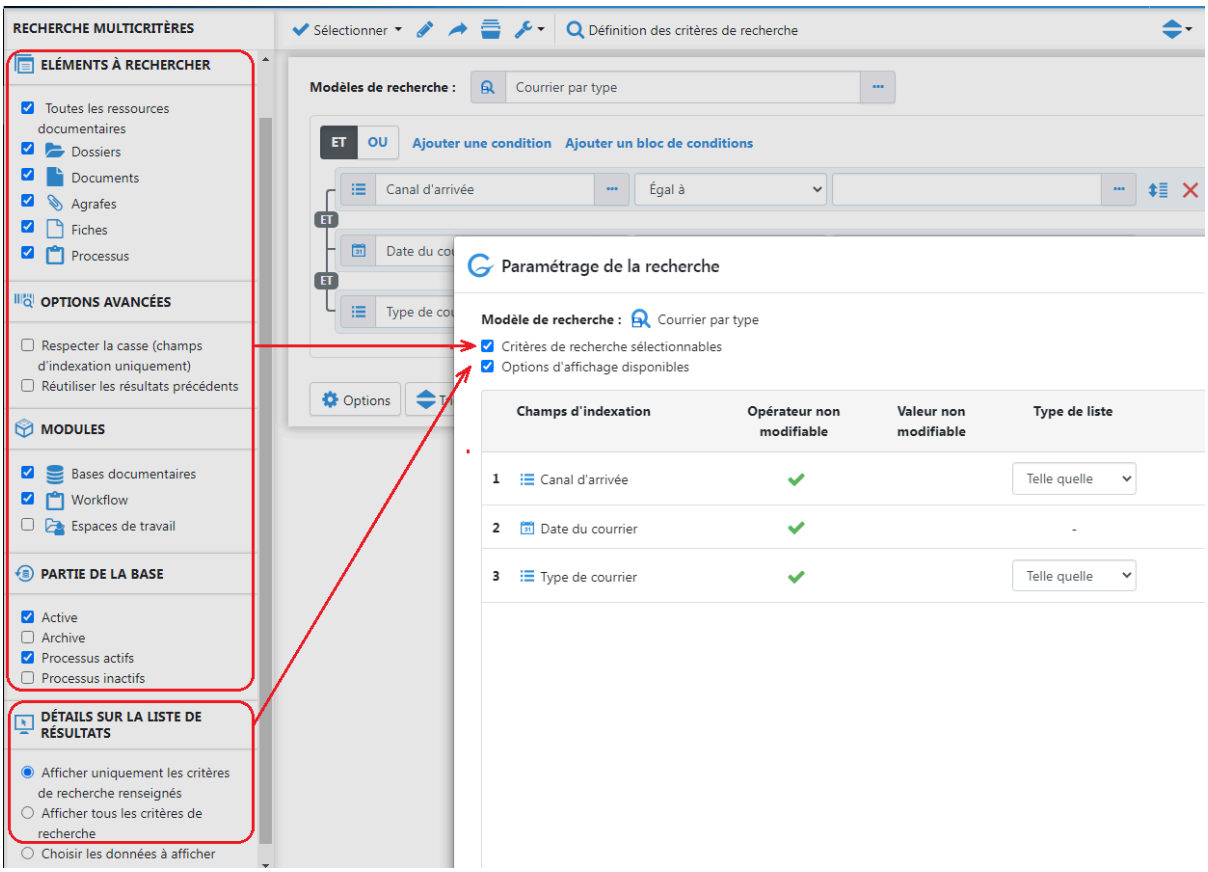
La version 8.3 enrichit les options d'opérateurs pour les critères alphanumériques en introduisant l'opérateur **Longueur égale à**.



Options à disposition pour les modèles de recherche sauvegardés

La version 8.3 enrichit les **Options** disponibles pour les modèles de recherche multicritères.

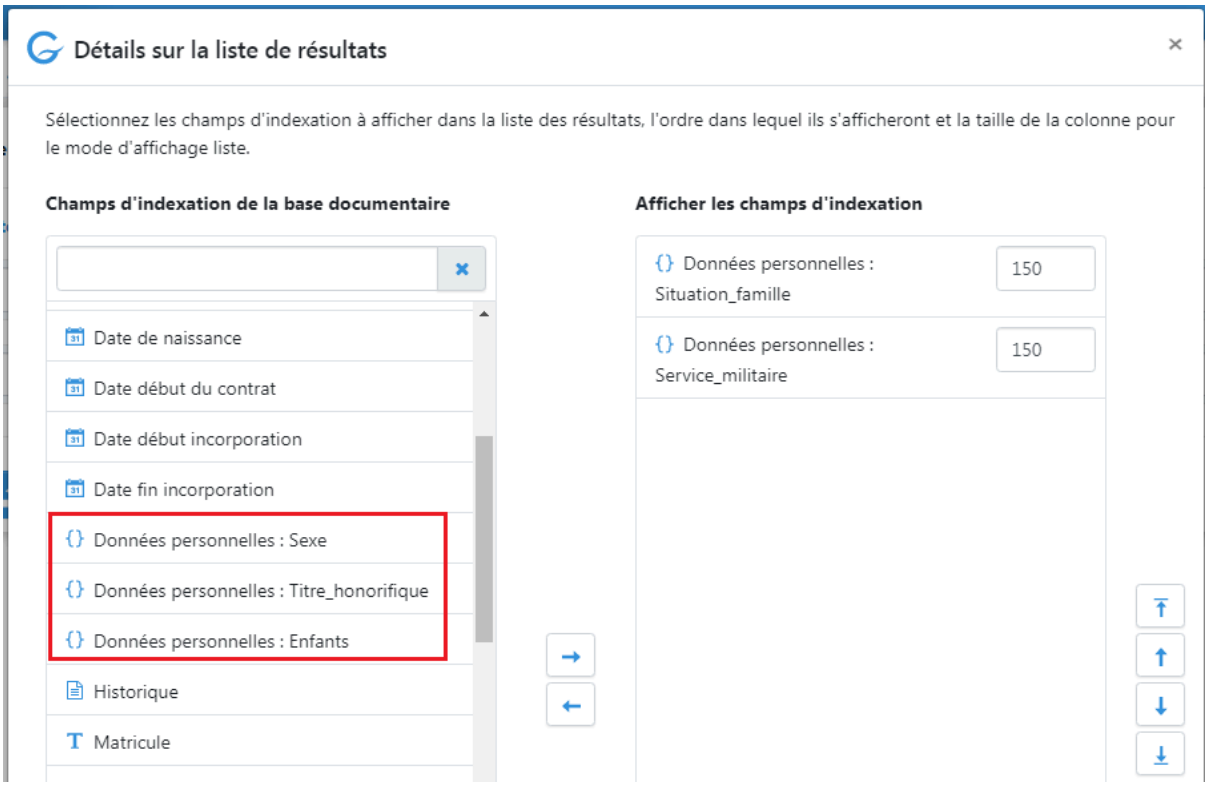
- **Critères de recherche sélectionnables** : permet d'autoriser l'utilisateur à modifier les critères de recherche.
- **Options d'affichage disponibles** : permet d'autoriser l'utilisateur à modifier les options d'affichage.



Par défaut, les deux options sont cochées.

Champs d'indexation à afficher dans la liste des résultats

La version 8.3 propose les champs d'indexation de l'entité pour personnaliser l'affichage de la liste des résultats.



Le champ d'indexation entité lui-même n'est pas disponible dans cette liste.

Module Tâches

Panneau d'affichage des documents

Basculement automatique du mode OCR au mode Aperçu

Lorsqu'un utilisateur traite une tâche configurée pour que le panneau d'affichage des documents soit en mode OCR et que la fonction d'OCR n'est pas disponible, l'affichage bascule automatiquement en mode Aperçu. Cette situation peut survenir si les extensions nécessaires ne sont pas installées ou si elles ne sont pas compatibles avec le dispositif utilisé, comme une tablette ou un téléphone.

Affichage des documents en plein écran

La commande **Basculer en plein écran** d'afficher les documents sur toute la largeur de l'interface, ce qui masque le formulaire de saisie. Pour revenir à l'affichage initial et faire réapparaître le formulaire de saisie, il suffit de cliquer sur le bouton **Quitter le plein écran,** qui se trouve au même emplacement.

Liste des tâches

Pagination de la liste des tâches

Par défaut, quand un utilisateur se positionne dans le module Tâches, Gargantua charge dans son interface la liste de toutes les tâches qu'il a à traiter, sans que l'utilisateur puisse interrompre ce chargement.

Lorsque les tâches à traiter sont nombreuses, il est préférable d'opter pour un chargement de cette liste par paquet de tâches (30 par 30 par exemple).

Pour activer ce mode d'affichage de la liste des tâches, il y a deux possibilités :

- Configurer la clé **-Dsiatel.config.inbox.listmode=paginated** dans le fichier catalina.bat de Tomcat avant le redémarrage du serveur,
- Définir cette clé dans l'interface Web, depuis l'url du serveur suivie de /toolbox, dans la page **Propriétés système** (accessible en tant qu'administrateur Gargantua).

Cette option de chargement par paquet ne s'applique pas aux brouillons car les nouveaux brouillons sont toujours ajoutés en fin de liste.

Le système de réactualisation de la liste des tâches à traiter a été adapté au mode d'affichage par paquet. Lorsque de nouvelles tâches sont à traiter, deux situations peuvent se produire selon que les paquets dans lesquelles elles doivent apparaitre sont déjà chargés ou non. Si le paquet est déjà chargé, alors la liste est réactualisée, sinon non.

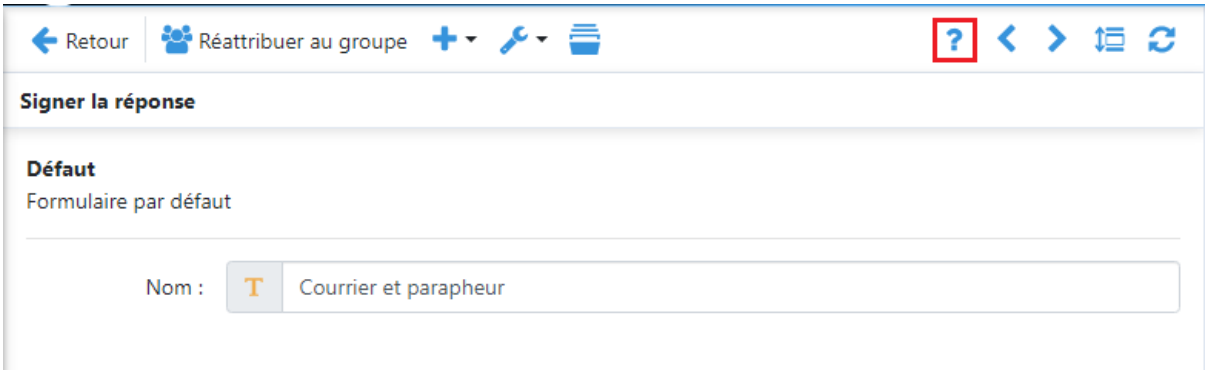
*Lorsqu'on utilise la commande « **Ouvrir l'emplacement/Accéder au document** » (soit suite à une recherche, soit via un lien reçu par email), le paquet contenant la tâche ainsi que tous les paquets précédents, sont chargés.*

Optimisation des temps d'affichage

La version 8.3 introduit des optimisations significatives qui accélèrent le temps d'affichage des listes de tâches, améliorant ainsi l'efficacité et la réactivité de l'interface utilisateur.

Aide en ligne relative à un processus

Si le paramètre **Référence de l'aide en ligne** est configuré au niveau d'un processus, une icône d'aide apparaît sur la barre de commande des tâches, permettant à l'utilisateur d'accéder à la page d'aide spécifiée.



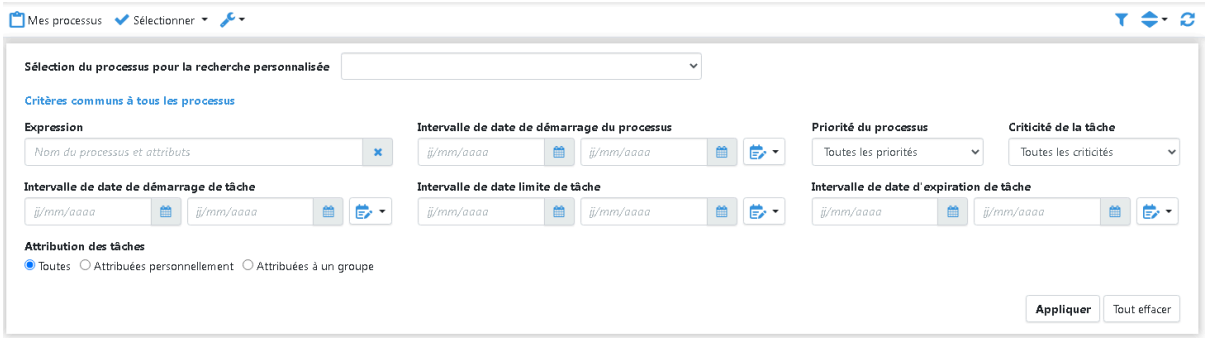
Panneau de recherche multicritères

Dans le bandeau de commandes de la liste des tâches, une icône **Filtre** permet d'ouvrir un panneau de recherche multicritères, destiné à aider les utilisateurs à retrouver rapidement les tâches qu'ils ont à traiter, en lien avec les critères qu'ils recherchent.

On trouve dans ce panneau deux catégories de filtres :

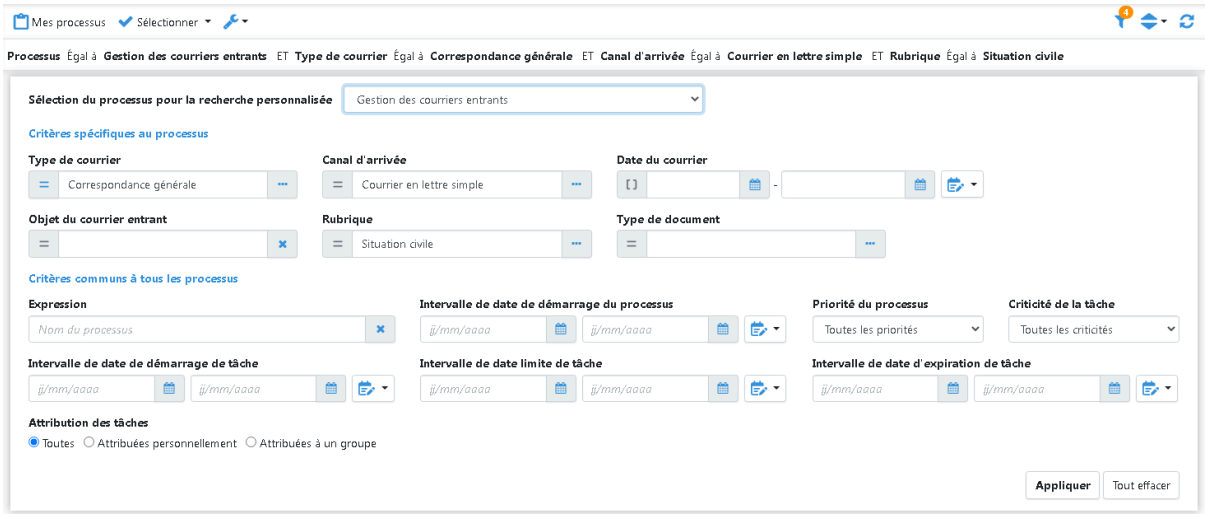
- Les filtres communs à tous les processus
- Les filtres spécifiques à un processus particulier, disponibles uniquement s'ils ont été configurés au préalable par l'administrateur.

Filtres communs



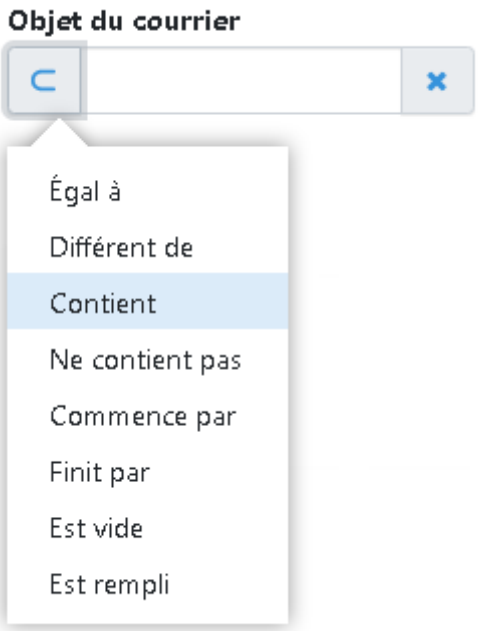
Filtres spécifiques à un processus en particulier

Quand l'administrateur a configuré des filtres spécifiques à un processus, une liste déroulante s'affiche dans le panneau. Celle-ci permet à l'utilisateur de choisir le processus pour lequel il désire effectuer une recherche personnalisée, en utilisant ces filtres spécifiques. Cette fonctionnalité facilite la réalisation de recherches ciblées en fonction des caractéristiques uniques de chaque processus.



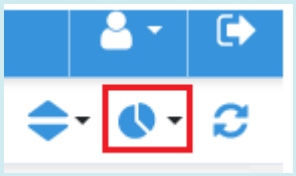
Choix des opérateurs

Chaque filtre est précédé de l'opérateur défini par l'administrateur. Visuellement, les opérateurs modifiables par l'utilisateur sont indiqués en bleu, tandis que ceux non modifiables sont en gris, facilitant ainsi leur identification. Pour modifier un opérateur, un clic suffit pour ouvrir un menu qui présente tous les opérateurs disponibles adaptés au type de filtre sélectionné.



Si le changement de l'opérateur n'est pas autorisé il est affiché en gris.

*Le bouton **Filtre par priorité** a été retiré, car il est devenu redondant, la fonctionnalité étant contenue dans les filtres communs.*



Option de filtres insensibles aux accents

Une option peut être activée pour que les filtres de recherche par expression dans les modules **Tâches** et **Supervision des processus** soient insensibles aux accents. Pour cela, il est nécessaire d'ajouter l'extension **unaccent** dans PostgreSQL en utilisant le compte de superutilisateur.

Module Acquisition de documents

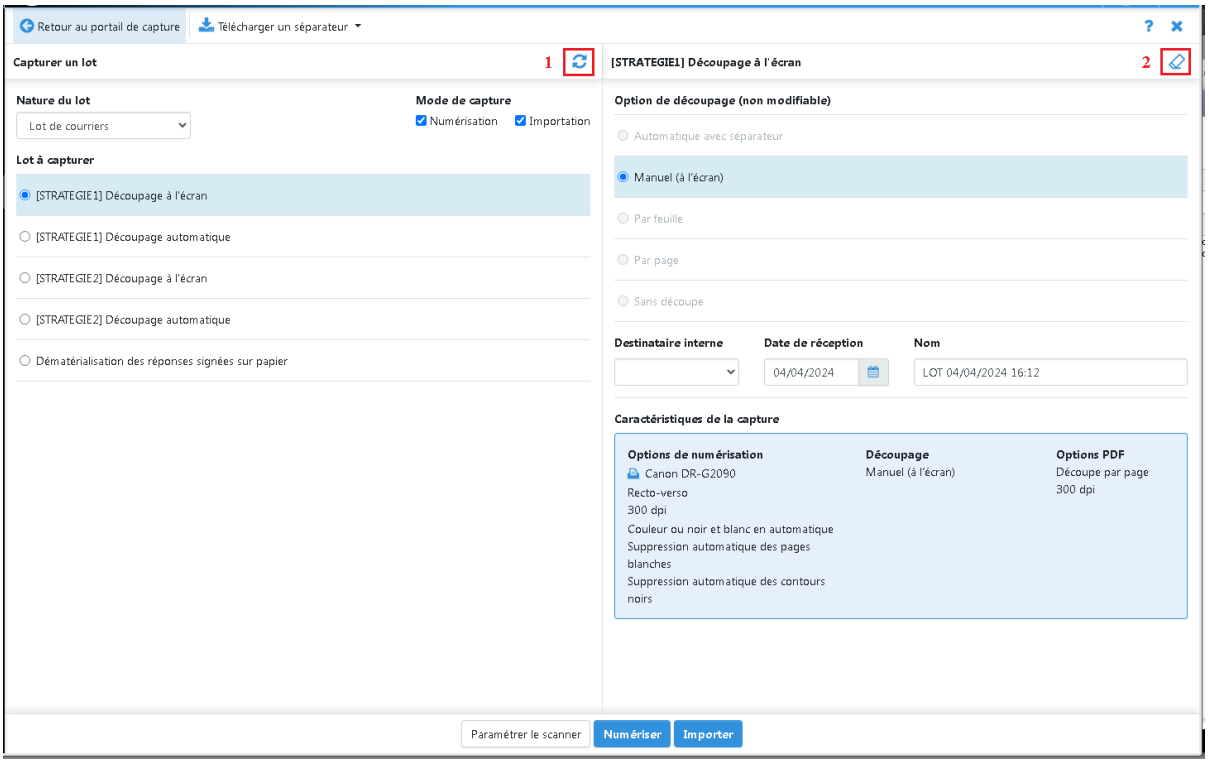
Etape Capture

La version 8.3 de Gargantua apporte de nombreuses améliorations et modernisations des interfaces. C'est ainsi le cas de l'étape de **Capture**.

Actualisation de la liste de lots et réinitialisation des paramètres par défaut

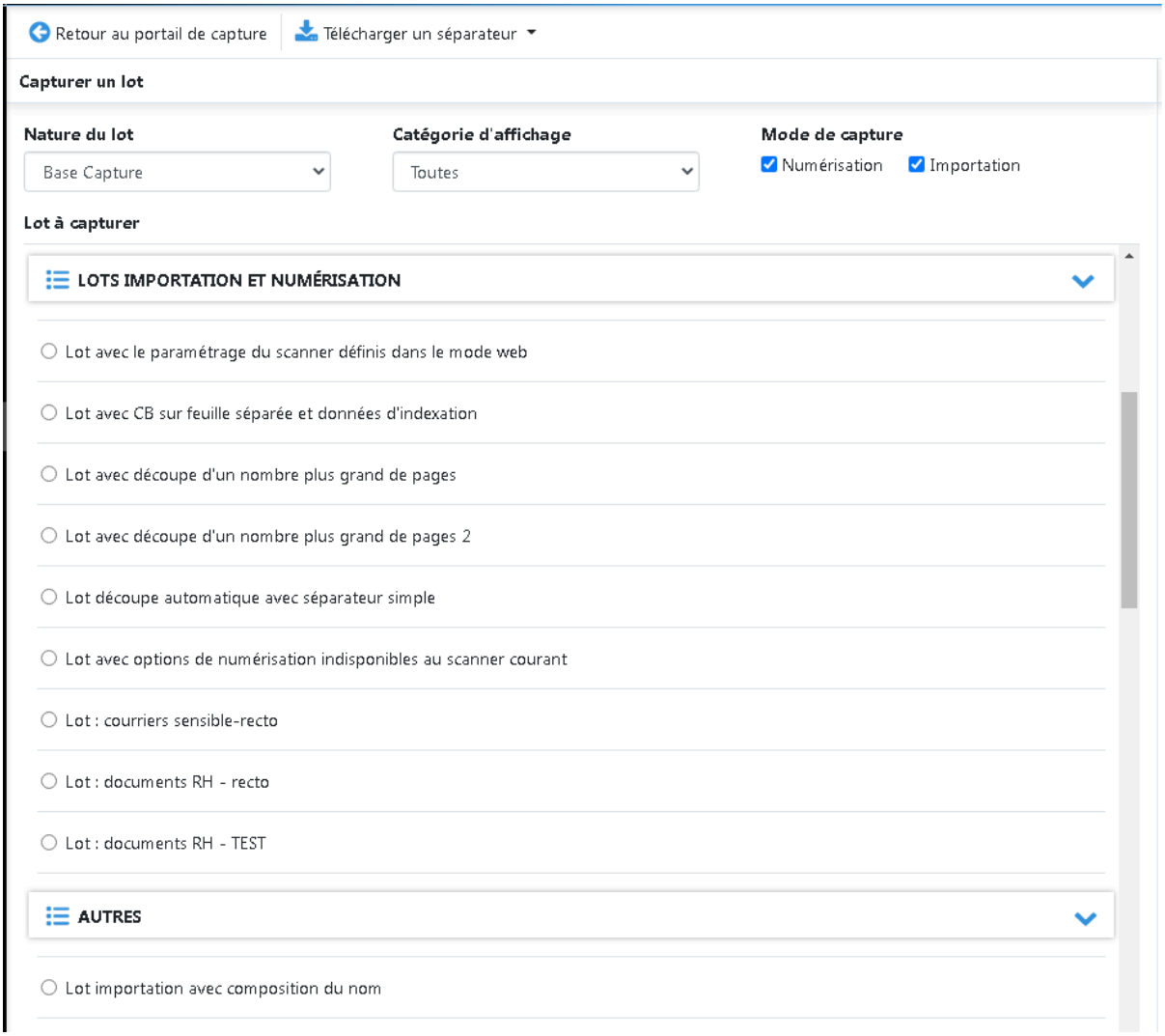
La commande **Actualiser** (1) permet de rafraichir la liste des lots.

Le bouton **Réinitialiser** (2) permet de restaurer les paramètres du lot qui auraient été modifiés par l'utilisateur, en revenant à leurs valeurs initiales.



Filtre des lots par catégorie

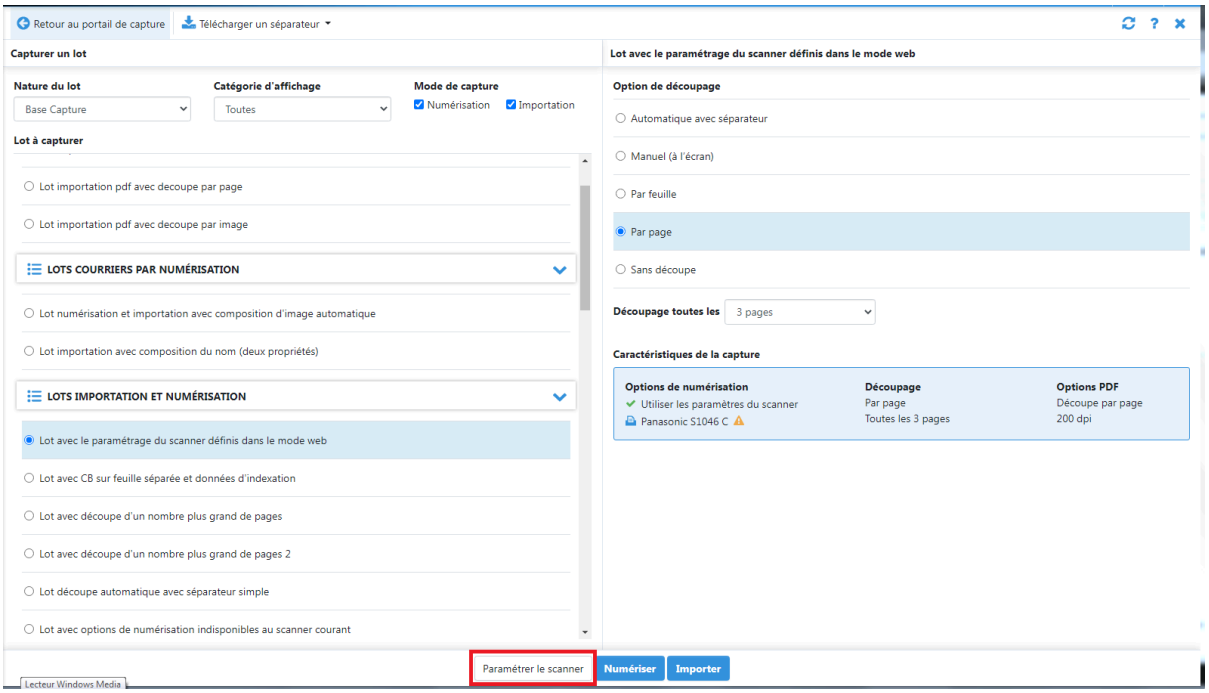
La liste déroulante **Catégorie** apparait à condition que votre Administrateur ait paramétré des regroupements de lots. En choisissant une catégorie spécifique, la liste se restreint pour afficher uniquement les lots associés à cette catégorie.



Les lots qui ne sont regroupés sous aucune catégorie sont affichés sous une catégorie générique **Autres**.

ACCÈS AU PARAMÉTRAGE DU SCANNER

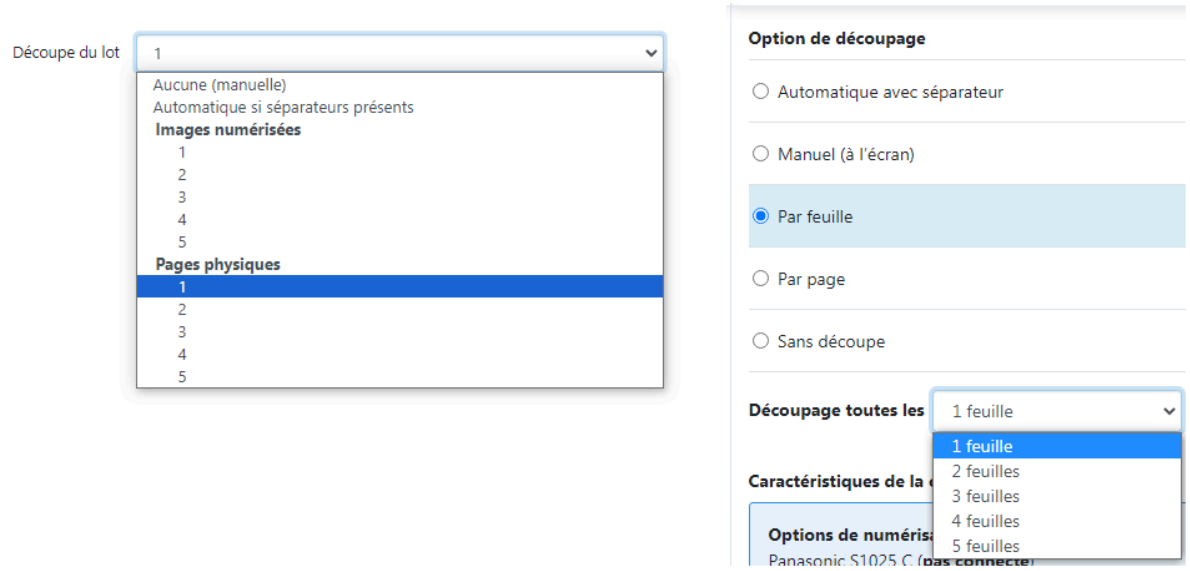
Pour les lots de documents ajoutés par **Numérisation**, le bouton permettant d'accéder au paramétrage du scanner est disponible sur l'écran de sélection du lot.



Les paramètres du scanner définis au niveau du lot sont toujours prioritaires, même si on les change dans le paramétrage du scanner. Toutes les modifications des autres paramètres du scanner sont prises en compte.

Modernisation des paramètres de découpe

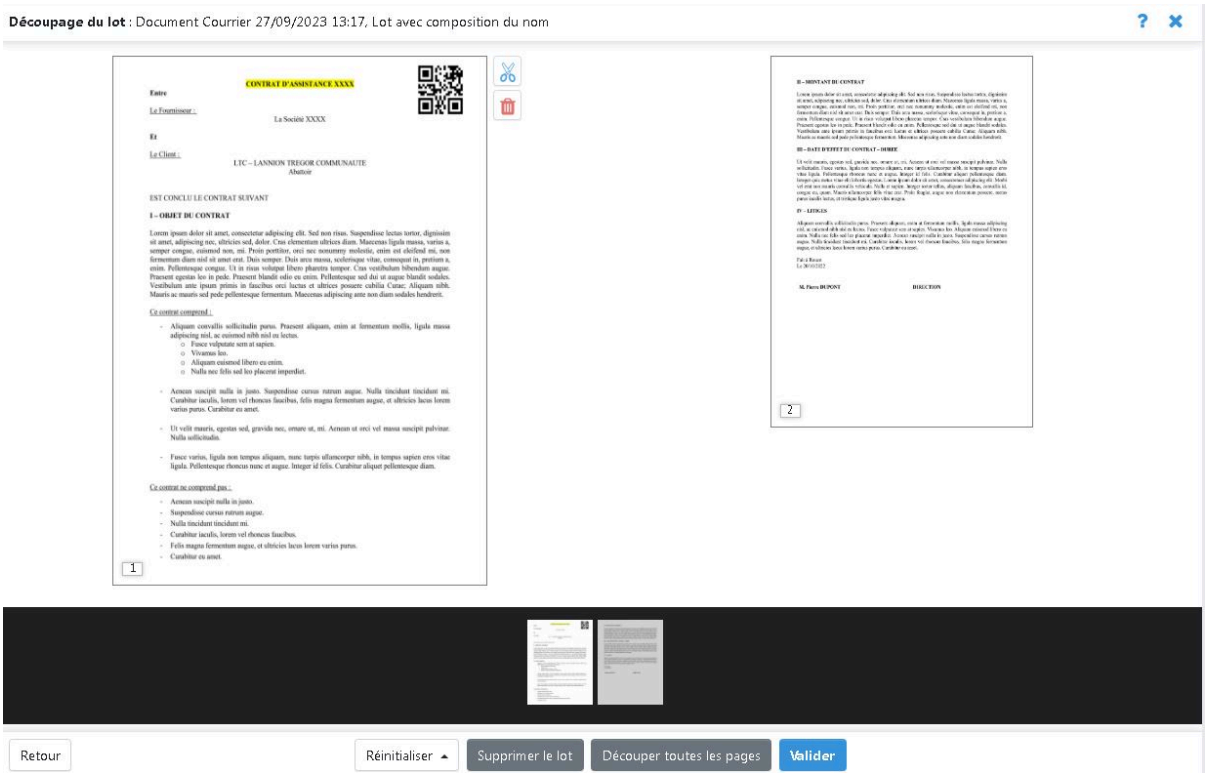
Les options de découpage du lot ont fait l'objet de modernisations. Cela est illustré par la figure ci-dessous. Dans la partie gauche, les anciens termes utilisés et à droite la nouvelle formule plus facile à comprendre car utilisant un langage naturel.



Pour les lots d'importation et les lots issus de numérisation uniquement recto ou uniquement verso, il n'est pas possible d'utiliser le mode de découpage à la feuille car seules des pages sont traitées.

ETAPE DÉCOUPE

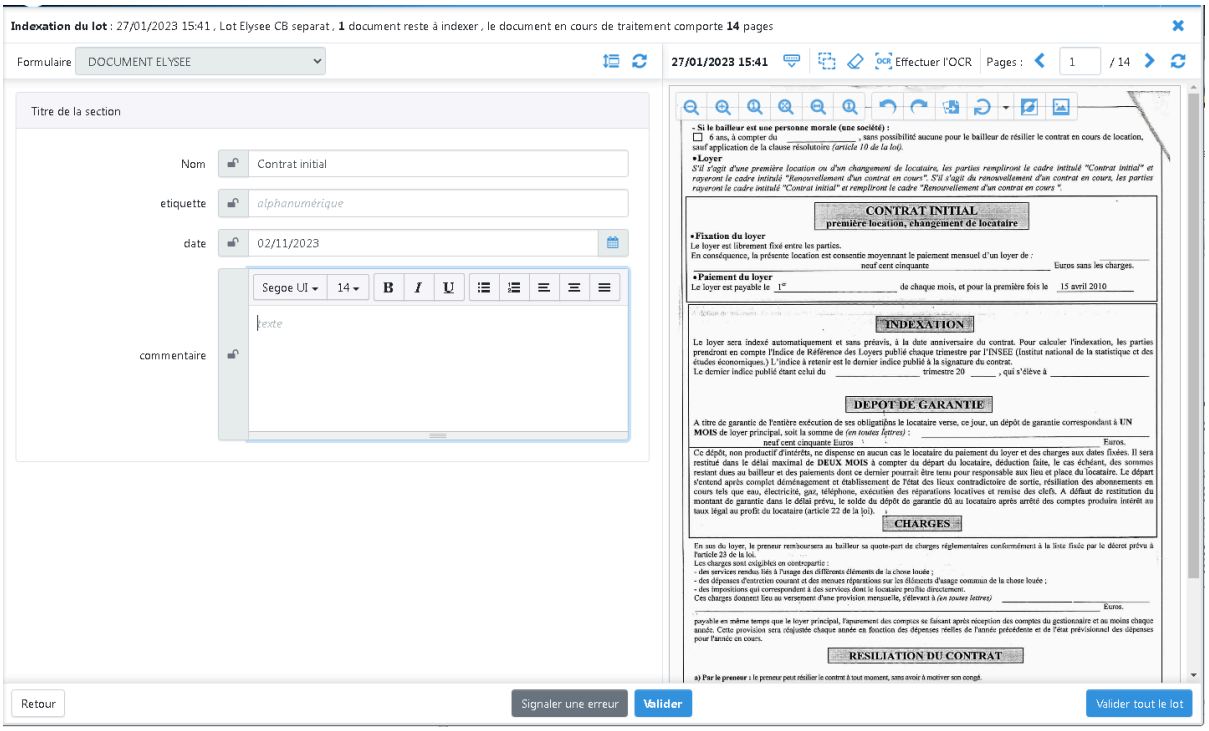
Dans l'interface de découpage du lot, l'ordre des boutons de commandes se présente comme illustré ci-dessous :



Le bouton **Retour** se situe à l'extrême gauche de l'écran.

Etape Indexation

Dans l'interface d'indexation du lot, l'ordre des boutons de commandes se présente comme illustré ci-dessous :



Pour rappel, le bouton **Valider tout le lot** ne s'affiche que dans les lots pour lesquels il est autorisé (dans la configuration du lot).

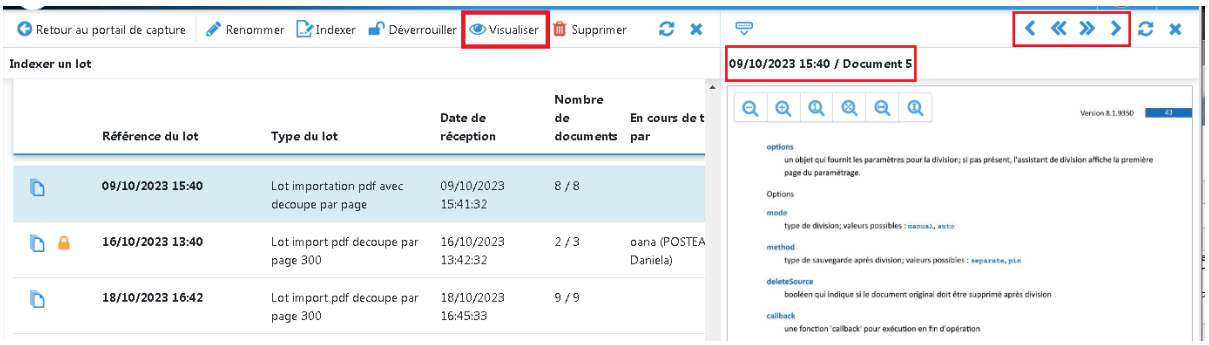
Le bouton **Retour** se situe à l'extrême gauche de l'écran.

Visualisation des documents d'un lot

Dans la liste des lots à indexer, la commande **Visualiser** permet d'ouvrir, dans le panneau de droite, l'aperçu de la première page de chaque document du lot.

Cette commande est accessible à partir de la barre d'outils située en haut de l'écran, et à partir du menu contextuel (qui s'ouvre par clic droit sur une ligne).

Dans le panneau **Aperçu**, deux zappeurs sont proposés, l'un pour visualiser les documents du lot (**<< >>**) et l'autre pour naviguer entre les lots de la liste (**< >**).



Dans ce panneau, le nom du lot est également renseigné, suivi du numéro d'ordre du document en cours d'affichage, dans le lot courant.

Aide en ligne

Un aide en ligne est disponible dans l'interface de chaque étape du module **Acquisition de documents**.

Des prérequis sont à respecter :

- Il faut que le plugin **siatel-plugin-onlinehelp.jar** soit déployé sur le serveur.
- Dans les options de configuration du module **Acquisition de documents** il faut avoir l'option **capture.help.reference** = CCAP

Module Carnet d'adresses

La version 8.3 amène un carnet d'adresses aux fonctionnalités considérablement enrichies par rapport au carnet d'adresses historique de Gargantua, avec notamment :

- Des contrôles de doublons a priori, a posteriori
- Des recherches approchées à l'aide d'un algorithme de Levenshtein
- Une ergonomie améliorée
- Des fonctions d'interfaçage avec la GED et le Workflow homogénéisées
- Liens relationnels entre les contacts physiques

Module Outils

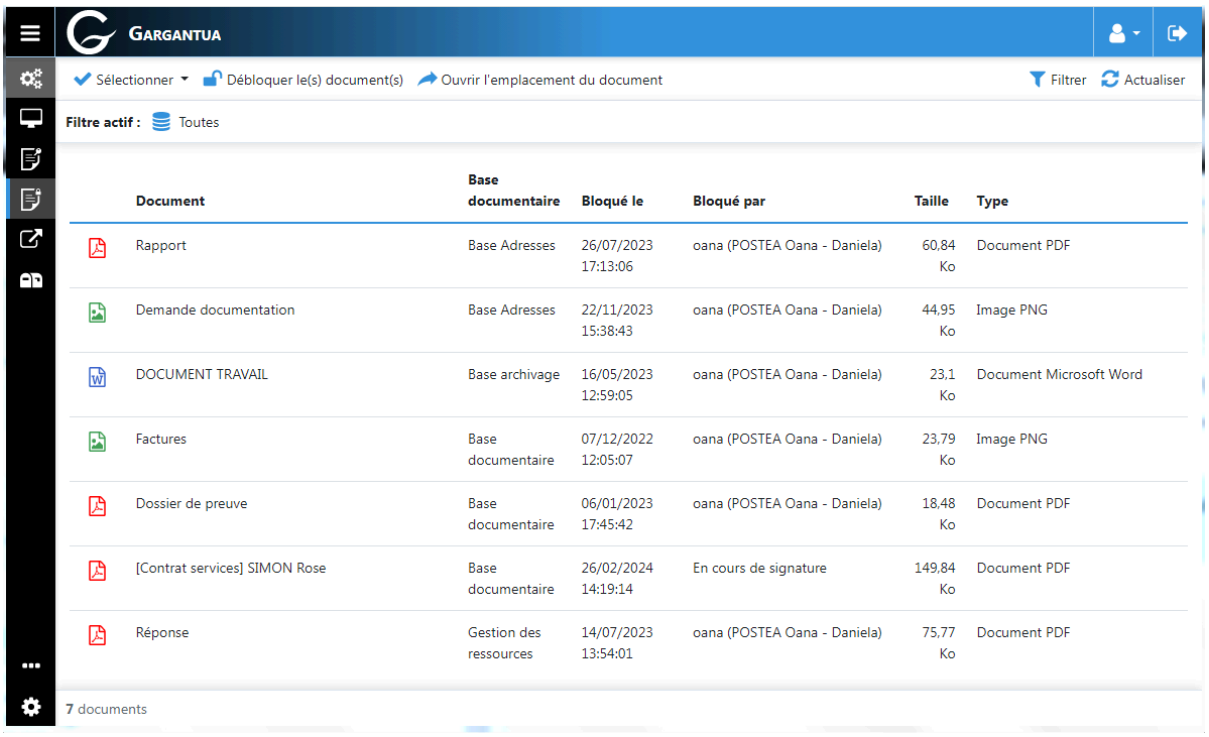
Gestion des documents bloqués

Les utilisateurs membres DBA ou gestionnaires ont accès au module dédié à la gestion des documents bloqués.

Ce module leur permet de :

- Débloquer des documents,
- Accéder à leur emplacement,
- Utiliser le panneau de recherche via l'icône Filtre pour retrouver rapidement les documents bloqués relatifs aux critères recherchés.

Les fonctionnalités offertes dans ce module sont similaires à celles qui se trouvent dans le module **Documents verrouillés**.



Gestion des liens de téléchargement

Depuis la version 8.3, la liste des liens de téléchargement créés présente :

- **Une icône PDF** pour repérer plus facilement les documents pour lesquels ce format de téléchargement a été retenu
- **L'information Sans expiration** lorsqu'aucune limite de date ou de délai n'a été retenue

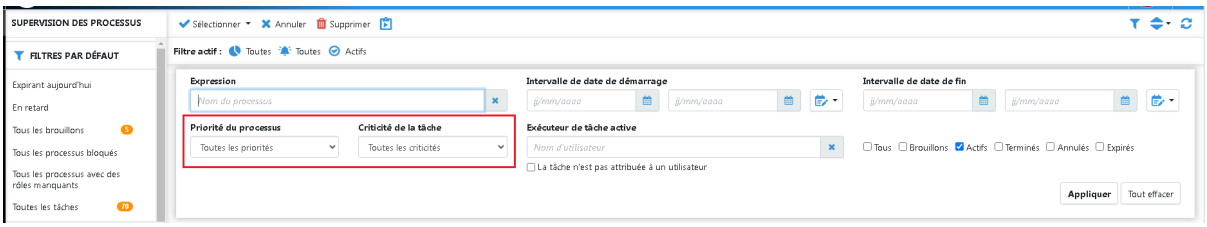
	Etat	Document	Base documentaire	Date d'expiration	Téléchargements disponibles	Lien	Utilisateur
	Actif	Cerfa 3	AGDREF	20/04/2024 23:59:59	3	http://192.168.1.64:8082/siatel-dew/app/linkto/NzNmS1psWS80djJxTTBXUjE4WmQ4MXJ6SVZxTmovbTFpRi83dnN6dG03blNiM0lVOERHdThoNWE0V1FKM3IrRFVTa2lKSjR5WTZ3PQ	oana (POSTEA Oana - Daniela)
	Actif	Cerfa 1	AGDREF	Sans expiration	Illimité	http://192.168.1.64:8082/siatel-dew/app/linkto/dzFnT3pDQ3RlVEd4NFJ6UlcrRFJ0KzNYSDdwdm4xanVTcFJlcFhDbC9uamNoK25kOXdUcVo1NWNCVEE3SXI1NnJxc1hJQWVJeVM0PQ	oana (POSTEA Oana - Daniela)
	Actif	1	Pour tous le monde	22/04/2024 17:09:59	Illimité	http://192.168.1.64:8082/siatel-dew/app/linkto/WVJrWDRwUi92L2tJZzJ4NDVQMDBuajZIT1pNcmIwSmxmTlhkWDFuZkpBTzJDR0dsdjQxRlFMMEQ5d3M2UUt6SlY2MmtzWmpnL2hRPQ	oana (POSTEA Oana - Daniela)
	Actif	1	Pour tous le monde	Sans expiration	Illimité	http://192.168.1.64:8082/siatel-dew/app/linkto/bUZFNncxZ1J5V3lCUVQ3TWx2ZEUwZVFTeUF5TFRyZGordk15U3dicUFlU3lKM1drSnR6VWtmYUt1SUxlVEtma0xzOW9WM3pKbkQ4PQ	oana (POSTEA Oana - Daniela)

Module Supervision des processus

Filtres de recherche disponibles

Dans le module **Supervision des processus**, la version 8.3 enrichit le panneau de recherche avec deux nouveaux filtres, permettant d'affiner la recherche selon :

- La priorité donnée à un processus,
- La criticité associée à une tâche.



Le filtre Criticité de la tâche s'applique uniquement aux tâches actives

Module Paramétrage

Rubrique Personnalisation des menus

Commandes communes aux dossiers et aux documents

Depuis la version 8.3, la commande **Envoyer vers** se trouve dans la section **Commandes communes aux dossiers**

et aux documents. Cette modification fait suite à une évolution qui rend désormais cette commande applicable aussi aux dossiers.

Pour rappel, cette commande se situait préalablement dans la section **Commandes spécifiques aux documents**.

Commandes spécifiques aux processus

La version 8.3 de Gargantua améliore la personnalisation du menu contextuel pour une tâche à traiter, permettant de choisir d'y afficher ou non les commandes **Annuler la tâche**, **Valider la tâche** et **Outils (journal)**.

Module Gestionnaire

Modifier le propriétaire

Une nouvelle commande a été ajouté dans la gestion des utilisateurs : **Changer le propriétaire**.

Elle permet de transmettre la propriété des objets d'un utilisateur vers un autre utilisateur.

Suggérer un mot de passe fort aléatoire

Dans la boite de dialogue de création d'un nouvel utilisateur, une option a fait son apparition : **Générer un mot de passe fort**.

Si cette option est cochée, un mot de passe est généré. L'adresse e-mail devient obligatoire et l'option d'envoi de mail de notification de création de compte se coche et devient non modifiable.

Le mot de passe fort généré contient 15 caractères, avec au minimum une majuscule, une minuscule, un chiffre et un caractère spécial.

Console d'administration

Paramètres du serveur

Plugins

Carnet d'adresses

La version 8.3 amène une refonte totale du carnet d'adresses, aux fonctionnalités considérablement enrichies par rapport au carnet d'adresses historique de Gargantua, avec notamment :

- Des contrôles de doublons a priori, a posteriori
- Des recherches approchées à l'aide d'un algorithme de Levenshtein
- Une ergonomie améliorée
- Des fonctions d'interfaçage avec la GED et le Workflow homogénéisées

Prérequis pour utiliser le carnet d'adresses

L'extension **Carnet d'adresses** utilise les extensions PostgreSQL **unaccent** et **fuzzystrmatch.**

Pour pouvoir créer ces extensions dans pgAdmin, il faut être connecté en superutilisateur.

Options de configuration

dataSource.url = jdbc:postgresql://<host>:<port>/<database>
dataSource.user = dummy
dataSource.pass = secret

builtin defaults
hibernate.dialect = <rendered> from dataSource.url for postgresql, oracle, sqlserver
hibernate.hbm2ddl.auto = update
javax.persistence.create-database-schemas = true

all « hibernate.* » and « javax.persistence.* » properties are forwarded to the JPA engine

string matching configuration
#
the matching algorithm to use: 'ilike' (default), 'levenshtein' or 'soundex'
matching.alg = ilike

```
#
# strength of the matching algorithm used: 'strict' (default), 'relaxed', 'loose', 'custom'
# applicable only for 'levenshtein' and 'soundex'
# matching.resolution = strict

# levenshtein customization
#
# matching.alg = levenshtein
# matching.resolution = custom
#
# insertion cost
# matching.levenshtein.ins_cost = 1
# deletion cost
# matching.levenshtein.del_cost = 1
# substitution cost
# matching.levenshtein.sub_cost = 1
# maximum distance for fast-fail
# matching.levenshtein.max_d = 1

# soundex customization
#
# matching.alg = soundex
# matching.resolution = custom

# matching.soundex.min_diff = 4

# alternate matching algorithm and options to use for search are
# the same as the above but prefixed with 'search.' and it will take precedence
# over the <no-prefix> equivalent option, but not over explicit call parameters

addressbook2.country.default = FRA
addressbook2.contact.code = {{id}}
addressbook2.show.contact.code = 1
addressbook2.contact.name.uppercase = 1
addressbook2.filter.bygroup = 0
```

Moteur de classement

Configuration d'un délai d'attente au démarrage de processus

SUCCESSIFS

Dans le fichier de configuration du module « Moteur de classement », il est possible de définir un délai d'attente entre chaque démarrage de processus successifs. Il a été mis en place pour éviter la génération de doublons.

Ce paramètre est à définir de la façon suivante :

autoimport.classification.[N].process.first.seconds = [nombre entier]

[**N**] représente la classification utilisée

[**nombre entier**] représente le délai souhaité en secondes

GESTION DES UTILISATEURS

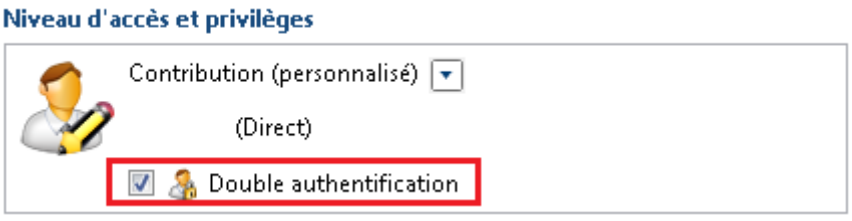
PRIVILÈGE « DOUBLE AUTHENTIFICATION »

La version 8.3 introduit le privilège **Double authentification** pour renforcer, si souhaité, le niveau de sécurité des connexions à Gargantua.

Lorsque ce privilège est activé, la connexion à l'interface utilisateur Gargantua nécessite une authentification à deux facteurs (ou double authentification) : après avoir saisi leur mot de passe, les utilisateurs doivent renseigner le code à usage unique et à durée limitée qu'ils reçoivent par SMS ou par mail, selon que leur numéro de mobile est renseigné ou non au niveau de leur compte utilisateur Gargantua. En l'absence de ces deux informations, la double authentification ne peut pas fonctionner.

Activation du privilège Double authentification

Ce privilège doit être renseigné soit directement au niveau des comptes utilisateurs, soit au niveau d'un de leurs groupes d'appartenance (auquel cas ils en héritent). Par défaut, les utilisateurs ne disposent pas de ce privilège.



Applications et utilisateurs concernés par la double authentification

Actuellement, la double authentification concerne uniquement la connexion à l'interface utilisateur Web. Elle s'applique aux utilisateurs internes de Gargantua ; les utilisateurs externes ne sont pas concernés par ce dispositif.

Prérequis pour l'envoi du code d'authentification par SMS

L'envoi du code d'authentification par SMS nécessite que le serveur soit configuré avec le module serveur **Envoi des SMS** (smsbridge).

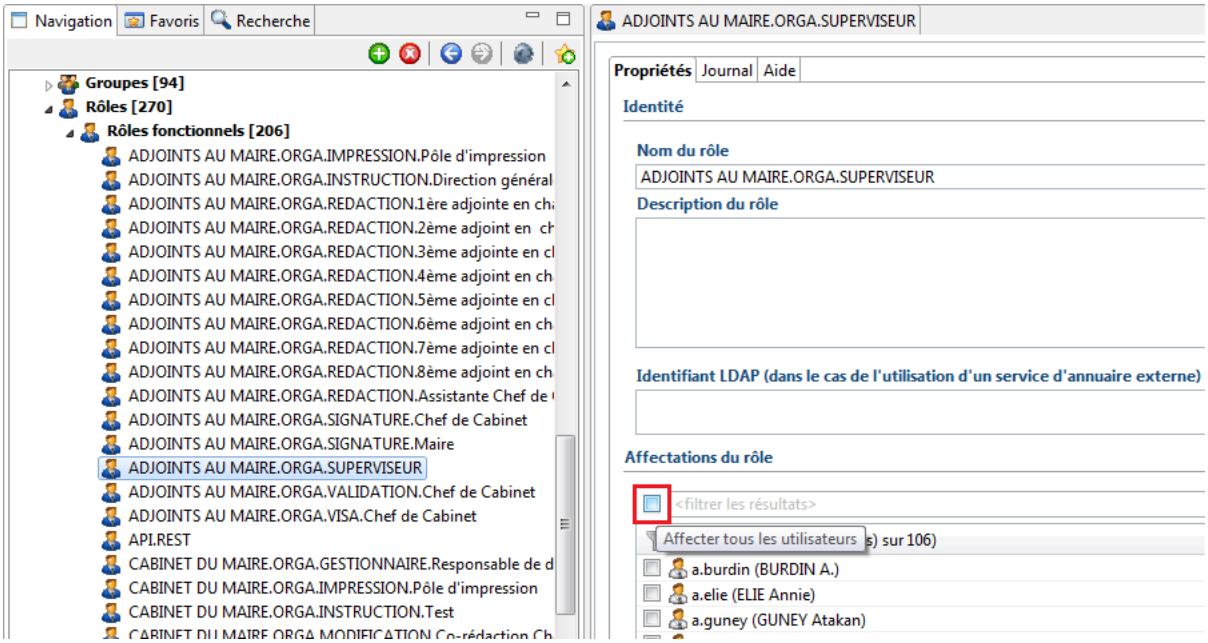
Principe de fonctionnement

1. L'utilisateur se connecte à l'interface Web.
2. Le serveur vérifie que l'utilisateur dispose du privilège **Double authentification**.
3. Si c'est le cas, le mode d'envoi du code d'authentification est déterminé :
    a. Si le numéro de mobile de l'utilisateur est renseigné et si le module serveur **Envoi des SMS** est configuré et activé, le serveur essaie d'envoyer le code par SMS. Si l'envoi du SMS échoue, le serveur applique le cas b).
    b. Si l'utilisateur dispose d'une adresse mail, le serveur envoie le code par mail.
4. L'utilisateur ne peut accéder à Gargantua tant que le code d'authentification n'est pas saisi. Le code est valide 10 minutes. L'utilisateur peut demander un nouveau code via l'interface utilisateur.

Gestion des rôles

Affectation en un clic de tous les utilisateurs à un rôle

Une case à cocher a été ajoutée dans le tableau des affectations d'utilisateurs aux rôles fonctionnels. Elle donne la possibilité de cocher tous les utilisateurs d'un coup pour les affecter à un rôle (fonctionnement identique à celui existant pour les groupes).

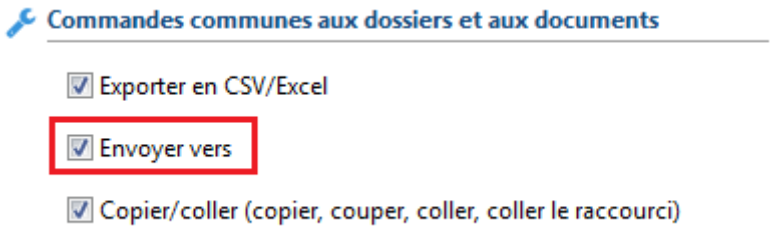


Personnalisation de l'interface par profil

Personnalisation des menus

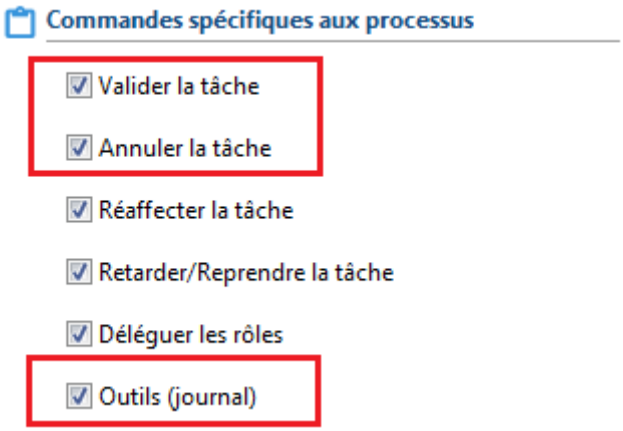
Commandes communes aux dossiers et aux documents

La commande **Envoyer vers,** initialement située dans la section **Commandes spécifiques aux documents**, a été transférée vers **Commandes communes aux dossiers et aux documents**. Cette modification fait suite à une évolution qui rend désormais cette commande applicable aussi aux dossiers.



Commandes spécifiques aux processus

La version 8.3 de Gargantua a amélioré la personnalisation du menu contextuel pour une tâche à traiter, permettant de choisir d'y afficher ou non les commandes **Annuler la tâche, Valider la tâche** et **Outils (journal)**.



Profils personnels

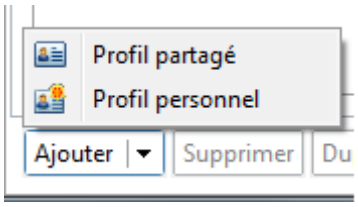
Création d'un profil personnel à partir d'un profil partagé

Avec l'introduction de la version 8.3, il est désormais possible de créer des **profils personnels**. Ces profils, spécifiques à chaque utilisateur, permettent une personnalisation individuelle de leur interface.

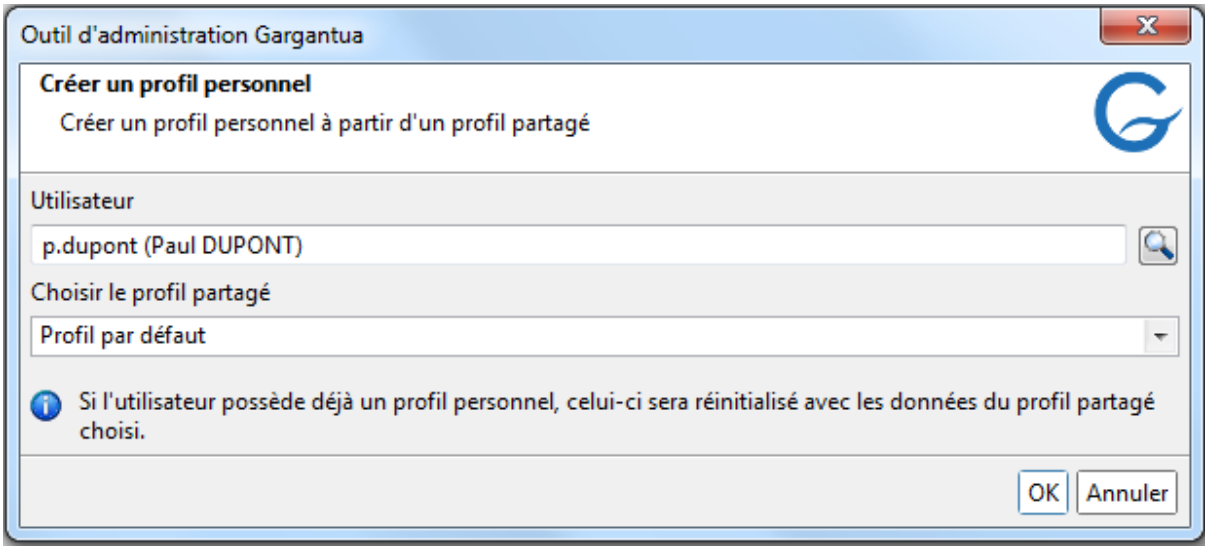
Pour rappel, jusqu'à la version 8.2, la création de profils depuis la console d'administration était limitée à des **profils partagés**.

Le bouton Ajouter, situé en bas à gauche du panneau de gestion de la personnalisation de l'interface par profil, propose deux options :

- **Ajouter un profil partagé**
- **Ajouter un profil personnel**



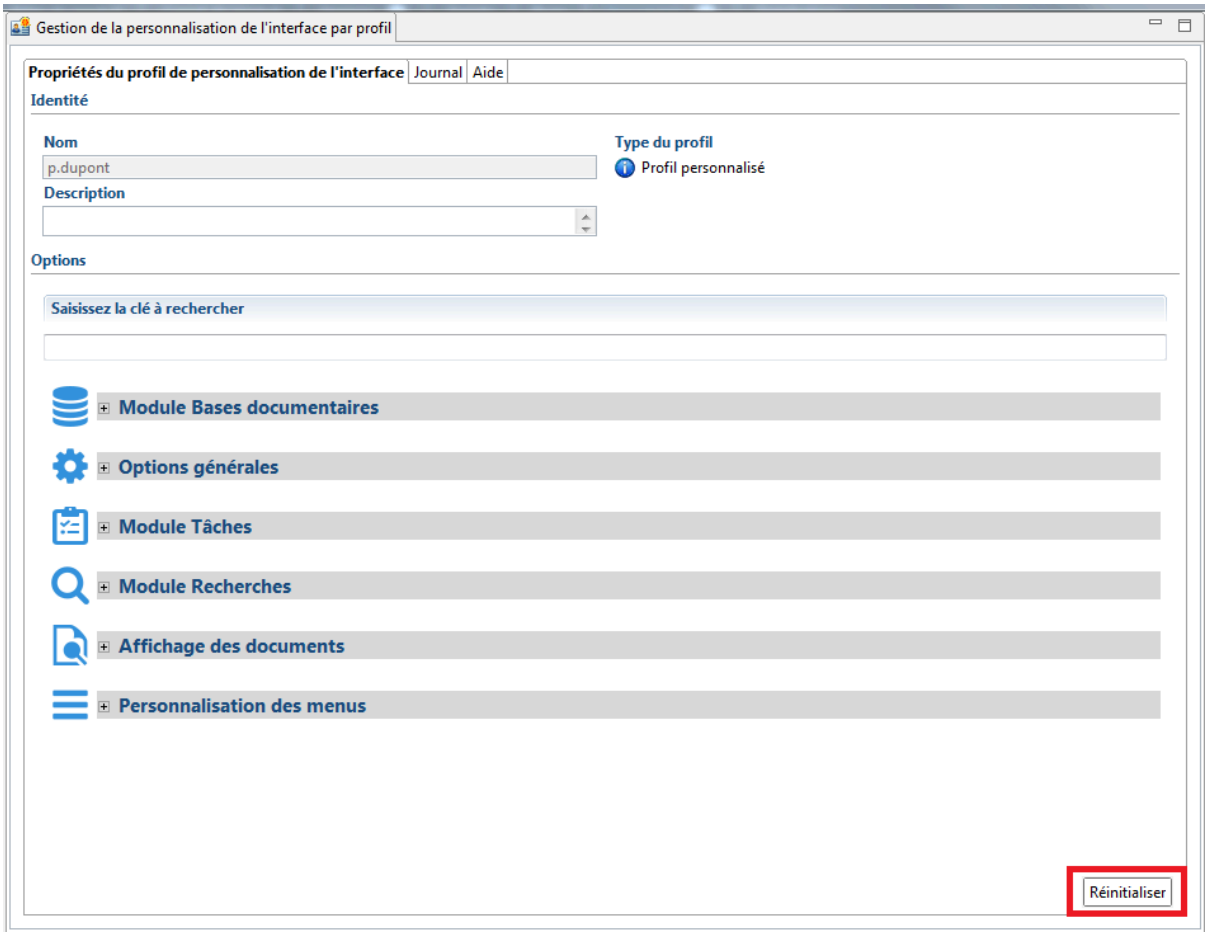
Lorsque l'option **Profil personnel** est retenue, une fenêtre permet de **désigner l'utilisateur** qui recevra le profil en cours de création. Cette interface permet également de **sélectionner un profil partagé** dont les données seront utilisées pour initialiser le nouveau profil personnel.



Si l'utilisateur possédait déjà un profil personnel, celui-ci sera réinitialisé avec les données du profil partagé choisi.

Réinitialisation d'un profil personnel avec les données d'un profil partagé

Depuis la version 8.3, la commande **Réinitialiser**, située en bas à droite du panneau d'affichage d'un profil personnel, permet de réinitialiser celui-ci avec les données d'un profil partagé.



Gestion des bases documentaires

Onglet Extensions

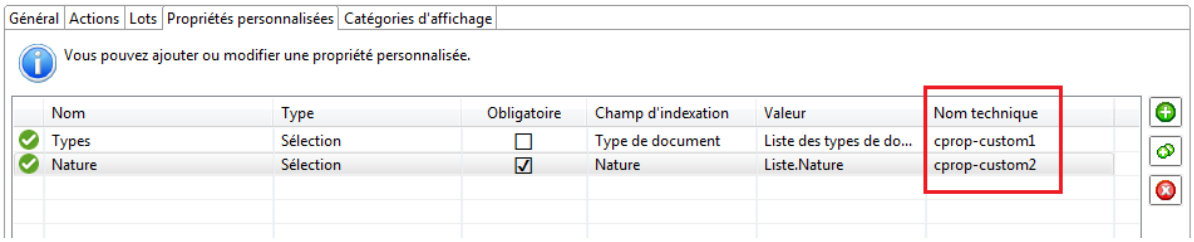
Extension/Module Acquisition de documents

Ce chapitre concerne la configuration du module **Acquisition de documents**, dans l'outil d'administration.

Propriétés personnalisées

Affichage du nom technique

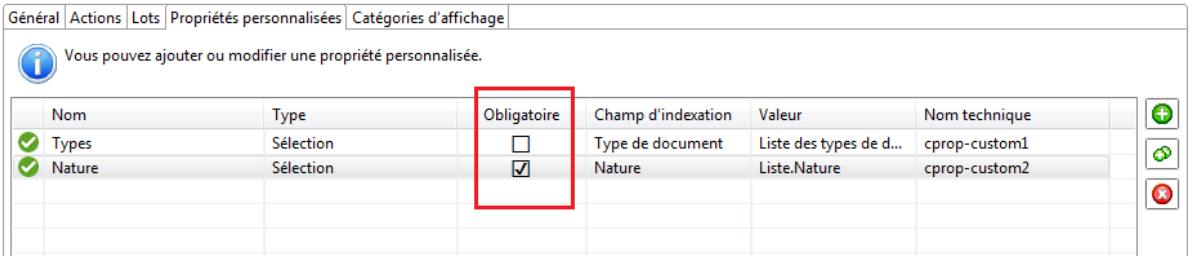
Dans le tableau de gestion des propriétés personnalisées, la dernière colonne permet d'afficher le nom technique de la propriété.



Ce nom technique peut être utilisé dans l'élaboration de l'expression servant à la composition du nom de lot (onglet Général).

Propriétés obligatoires

Avec la version 8.3 de Gargantua, il est maintenant possible de spécifier si une propriété personnalisée doit être obligatoire ou non.



Filtrage de champs d'indexation en fonction du formulaire de la cible

Jusqu'à présent, dans la classification pour la cible de type GED et dans les listes de champs disponibles pour OCR dans les lots avec cible GED ou processus, tous les attributs du formulaire s'affichaient, sans prendre en compte la page du formulaire définie pour l'affichage dans l'étape d'indexation.

Maintenant :

- dans la classification pour la cible GED : s'affichent les champs d'indexation de la page défaut du formulaire établie pour indexation (y compris les champs d'indexation cachés)

- dans la liste de champs d'indexations disponibles pour OCR dans un lot avec cible GED : s'affichent les champs d'indexation de la page défaut du formulaire de la cible. Les types de champs d'indexation pour lesquels la propriété OCR (dans le formulaire) n'est pas disponible sont exclus de la liste.
- dans la liste de champs d'indexation disponibles pour OCR dans un lot avec cible Processus : s'affichent les champs d'indexation de la page établie comme défaut dans le formulaire de la tâche de démarrage choisie dans la cible processus

Le filtrage des champs d'indexation s'applique également dans le paramétrage du lot à la liste des champs pouvant être choisis comme destination pour le texte récupéré des codes-barres avec données d'indexation et de l'imprimante.

Paramétrage d'un lot

Options de numérisation

Dans les options de numérisation par lot, deux options ont été exclues : recto-verso juxtaposées et recto-verso consécutives.

Les options suivantes sont disponibles : recto, verso et recto-verso.

Les types de compression **PDF Files** et **PDF multi pages** ne sont plus disponibles dans le paramétrage du lot.

Traitement du fichier PDF

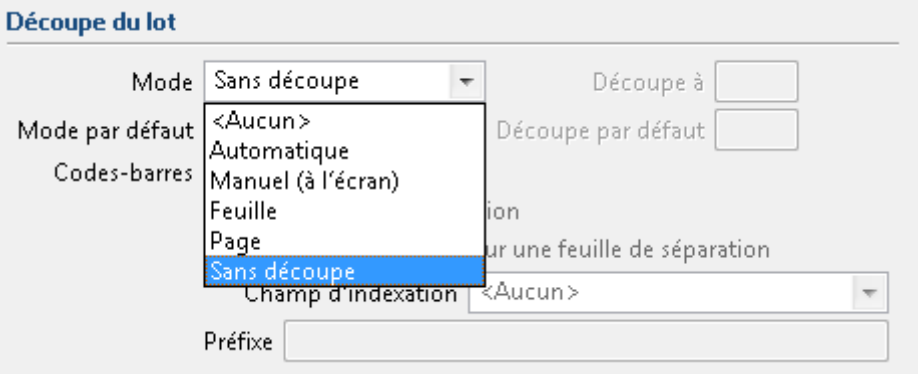
Désormais, ces options concernent uniquement les fichiers importés.

En conséquence, elles seront disponibles dans les lots avec type de capture **Importation**.

Options de découpage

Mode « Sans découpe »

La version 8.3 amène le mode de découpage **Sans découpe**.



Si ce mode est sélectionné, toute feuille séparatrice dans un lot est ignorée.

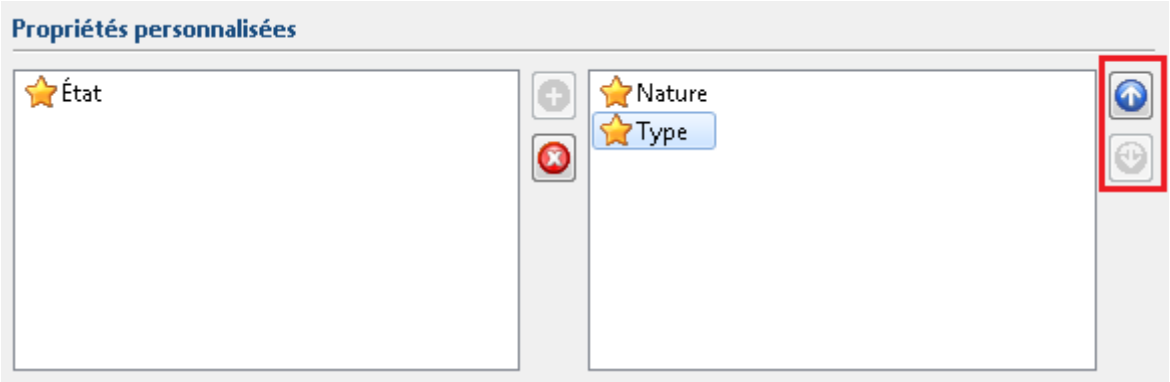
Mode « Découpage automatique » - Code barre et report d'une donnée dans un champ de type liste

Avec la version 8.3, lorsqu'un code barre est porteur d'une information, celle-ci peut être reportée dans un champ d'indexation de type liste.

Ce champ d'indexation doit figurer dans le formulaire défini pour la cible du lot.

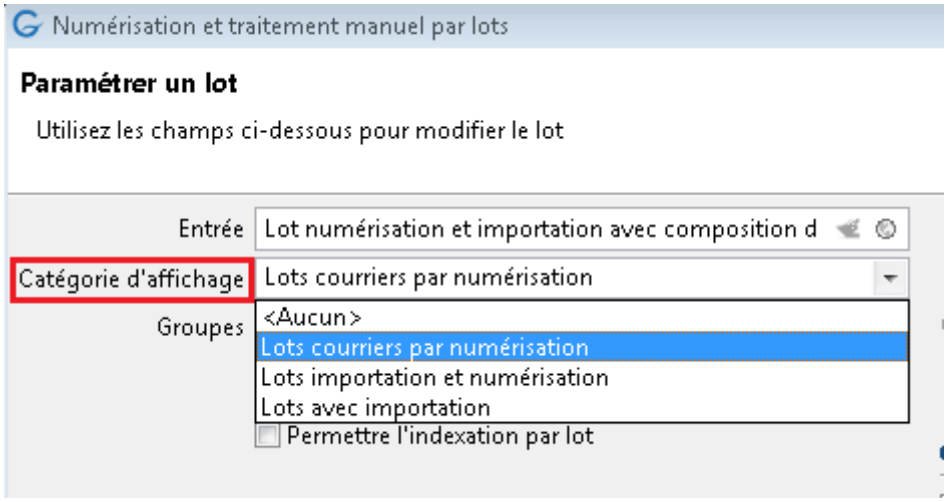
Ordre des propriétés personnalisées

Les boutons de modification de l'ordre des propriétés sont présents sur l'interface d'ajout de propriétés personnalisées.



Grouper les lots en catégories

L'écran de paramétrage du lot offre la possibilité de grouper par **Catégorie d'affichage**.



Toutes les catégories créées dans l'onglet **Catégories d'affichage** sont affichées dans cette liste.

Commandes désactivées dans l'interface utilisateur

Les commandes **Télécharger PDF** et **Conversion** ne sont plus disponibles ni dans le paramétrage général du module **Acquisition de documents**, ni dans le paramétrage du lot.

Désormais, ces options sont automatiquement et toujours désactivées (bloquées) au niveau de l'interface utilisateur.

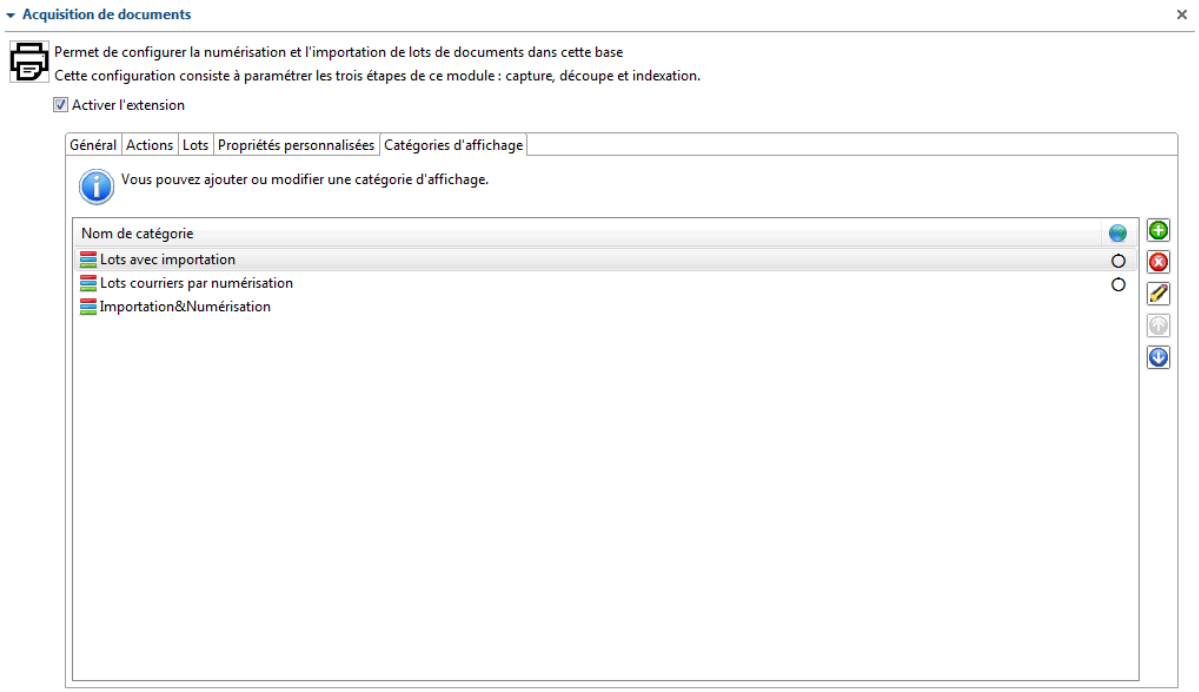
Saisie des catégories d'affichage

Avec la version 8.3 de Gargantua, l'onglet **Catégories d'affichage** fait son apparition.

Il permet l'ajout des catégories qui sont utilisées pour regrouper les lots lors de l'affichage dans l'étape de **Capture.**

Les noms des catégories peuvent être traduits.

L'ordre dans lequel les catégories s'affichent dans l'interface utilisateur peut également être défini.



Disponibilité de la fonction form_viewDocument

La fonction **form_viewDocument**, qui permet l'affichage d'un document dans la page d'indexation d'un formulaire est maintenant disponible dans le module capture.

L'appelle de la fonction : `form_viewDocument(docId)`

« docId » correspond à l'identifiant d'un document.

Module Workflex

Pour rappel, le module Workflex permet, depuis l'interface utilisateur, la gestion des bibliothèques de circuits, ainsi que la gestion des rôles désignés pour agir à une étape donnée du circuit.

Le nom de ce module est personnalisable à l'aide de la clé de configuration **workflex.label.**

Clé de configuration pour afficher ou non les rôles affectés à un utilisateur

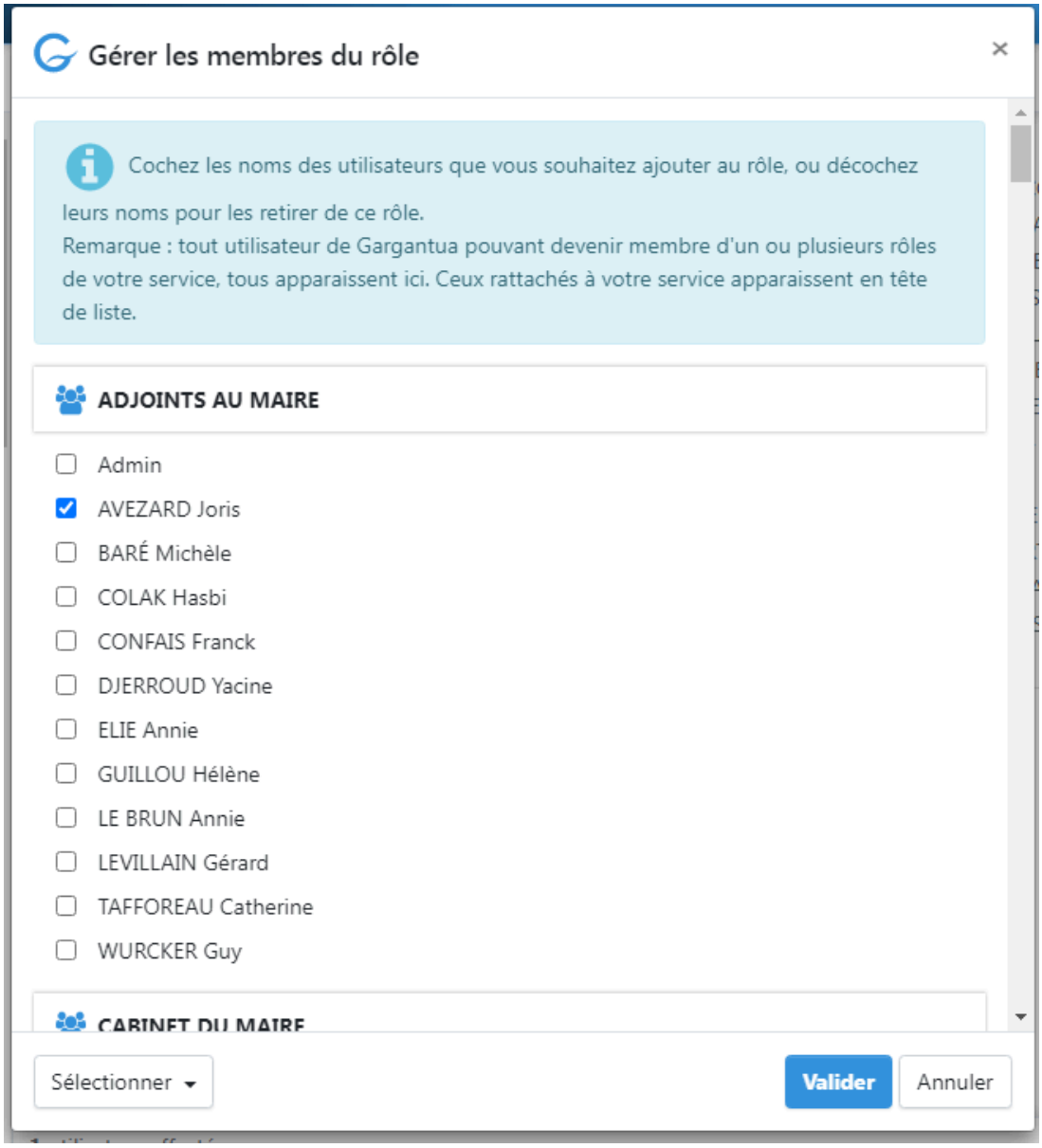
Depuis la version 8.3, une clé **workflex.assignments.display.mode** permet d'indiquer si les rôles affectés à un utilisateur doivent être affichés ou non dans la boîte de dialogue **Gérer les membres du rôle.**

workflex.assignments.display.mode = compact

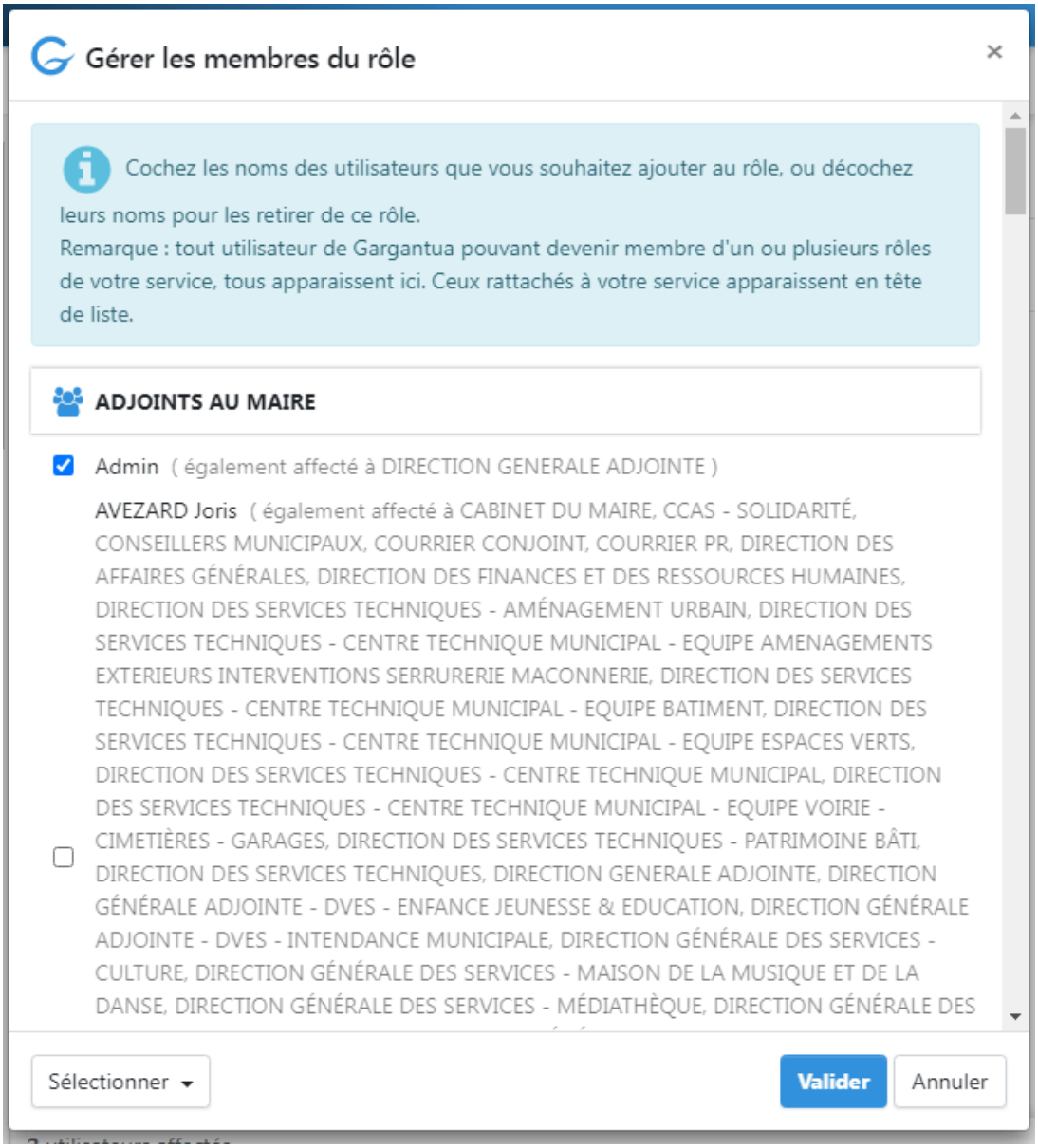
Deux valeurs sont possibles : **compact** (par défaut) / **detailed**

Voici l'affichage obtenu selon la valeur choisie :

- **workflex.assignments.display.mode** = detailed



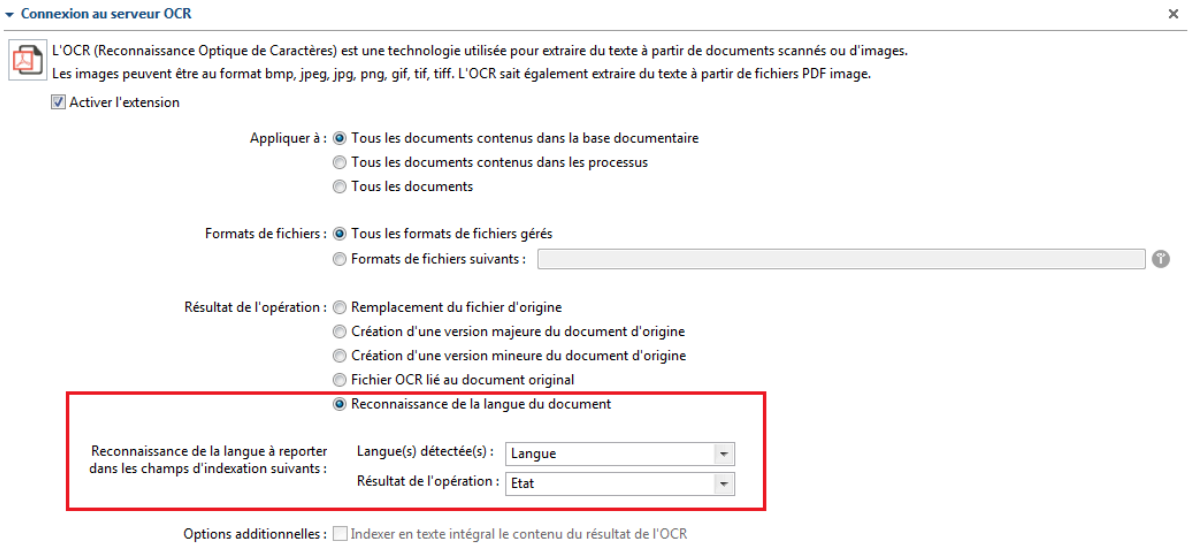
- **workflex.assignments.display.mode** = compact



Extension Connexion au serveur OCR

Reconnaissance de la langue du document

La version 8.3 propose un nouveau type de résultat pour l'opération d'OCR : **Reconnaissance de la langue du document**.



Ce mode de fonctionnement permet de retourner uniquement la langue du document (sans extraire le texte du document). Seuls les champs Langue(s) détectée(s) et Résultat de l'opération.

Pour ces deux champs, seules sont acceptées des valeurs de type alphanumériques.

Le résultat de la détection de la langue est au format JSON. Le contenu du résultat JSON de l'opération de détection est placé dans le contenu du champ d'indexation spécifié.

Les valeurs possibles pour le champ **Résultat de l'opération** sont :

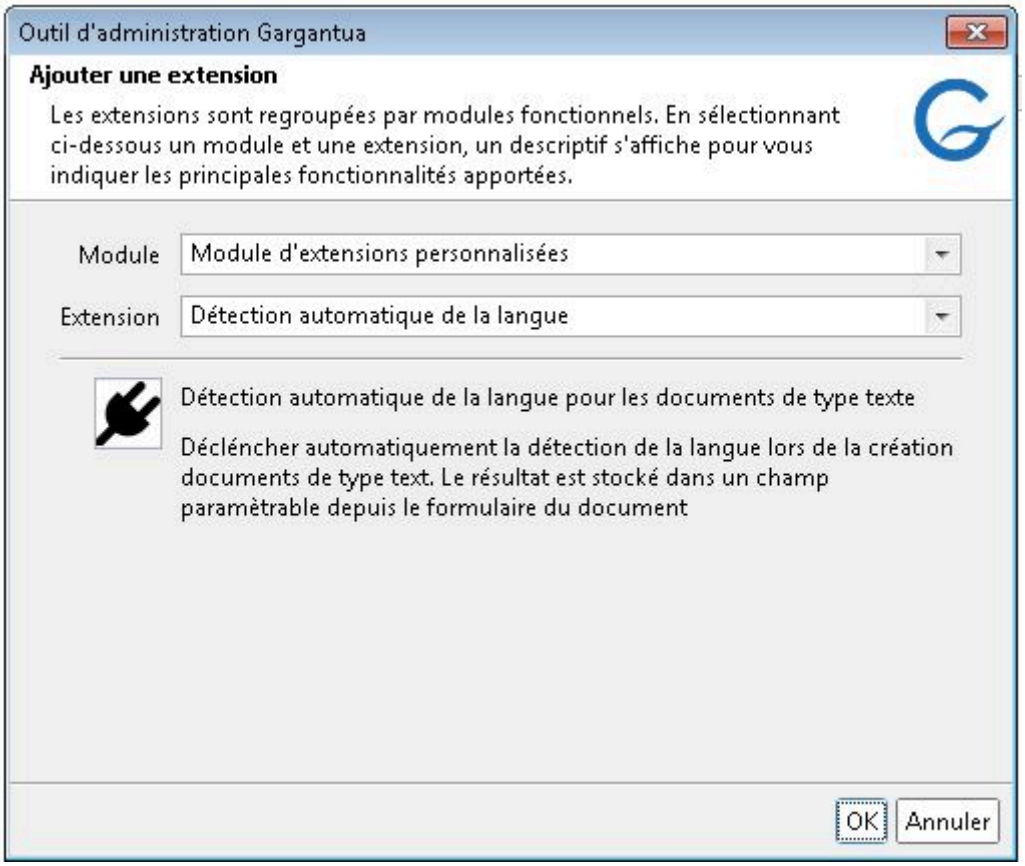
- **ignored** : ceci indique que le type de document n'est pas géré par le service OCR (c'est un fichier '.txt' par exemple) ou le domaine de document n'est pas permis (le document c'est en GED et la configuration indique seulement WF, ou vice versa)
- **canceled** : le document est un pdf qui contient déjà du texte ou il est signé et dans ce cas on n'y touche pas.
- **pending** : le document n'est pas encore traité, habituellement dans la file d'attente de traitement ou en cours de transfert vers le service OCR.
- **processing** : contenu transféré vers le service OCR, en attente d'un résultat
- **success** : opération terminée avec succès
- **skipped** : document ignoré par le service OCR (texte détecté dans le contenu, du côté du service OCR)

- **error** : erreur de détection ou de communication avec le service OCR
- **timeout** : l'opération n'a pas abouti dans le délai d'attente configuré

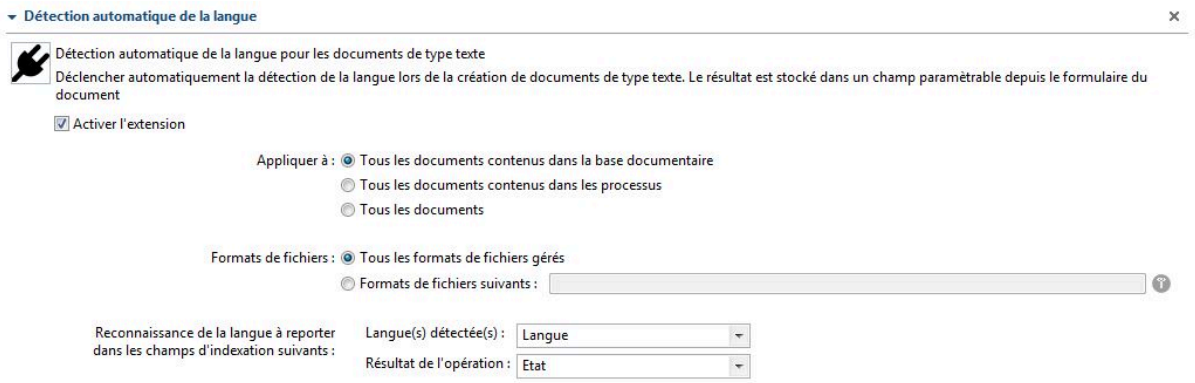
A voir aussi le chapitre **Méthodes de détection de la langue** dans le chapitre **Serveur OCR**.

Extension Détection automatique de la langue

Une nouvelle extension **Détection de la langue du texte** est disponible en base documentaire.



La configuration de l'extension donne plusieurs possibilités d'application de la détection, soit pour tous les documents, soit pour certains types de fichiers texte et d'obtenir l'état de l'opération et le résultat de la détection de la langue dans un attribut indiqué.



Apache OpenNLP est intégré comme moteur de détection de la langue.

Le modèle de langages utilisé est téléchargé depuis : https://opennlp.apache.org/models.html , c'est le fichier « langdetect-183.bin ». Dans le document README lié a ce fichier on peut lire « **The model can detect 103 languages in ISO 639-3 standard** » et dedans il y a aussi la liste avec les langages connus : https://downloads.apache.org/opennlp/models/langdetect/1.8.3/README.txt

> ***Le modèle Apache OpenNLP utilisé détecte 103 langues selon la norme ISO 693-3 et fonctionne bien avec des textes plus longs contenant au moins 2 phrases ou plus dans la même langue.***

Gestion des processus

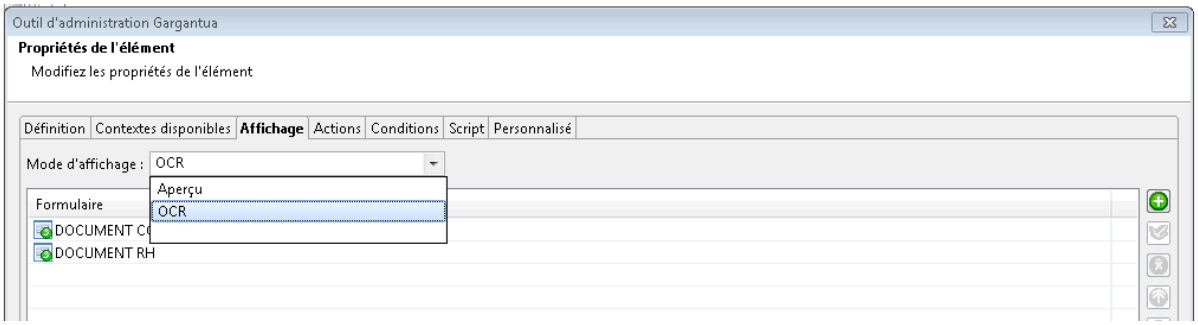
Mode d'affichage du panneau Documents dans une tâche de démarrage (Aperçu ou OCR)

Au niveau de la configuration d'une tâche de démarrage de processus, il est possible de définir comment afficher les documents dans le panneau prévu à cet effet. Deux options sont possibles :

- **Aperçu**, qui permet l'affichage du document,

- **OCR**, qui permet d'utiliser les fonctions d'OCR par encadrement de zones.

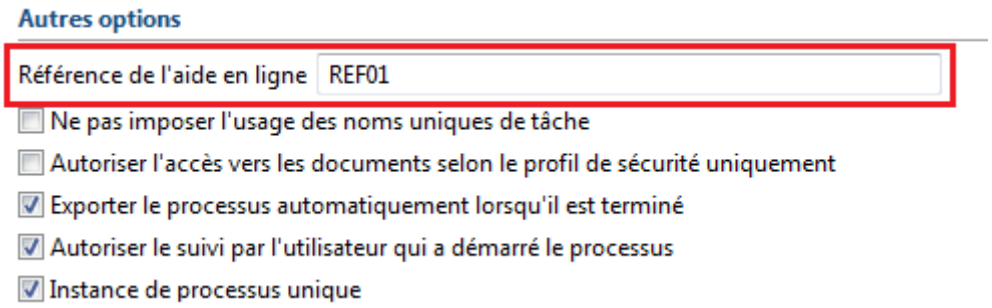
Quel que soit le mode d'affichage retenu (**Aperçu** ou **OCR**), il convient de préciser quels documents sont à afficher dans le panneau d'affichage. Ceci se fait en dressant la liste des formulaires qui caractérisent ces documents.



Aide en ligne

Il est possible d'associer une documentation à un processus sous forme d'une aide en ligne. Celle-ci doit nécessairement être rédigée par les équipes de Siatel à l'aide d'un outil interne de génération d'aide en ligne.

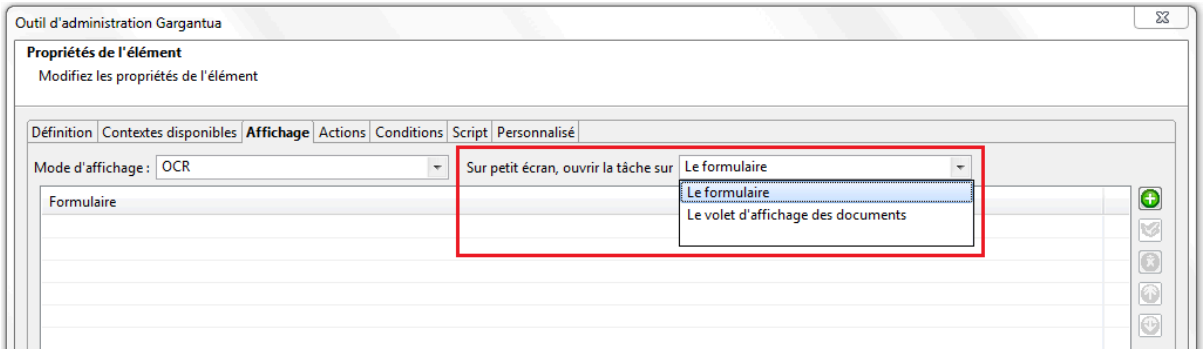
Pour faire apparaître l'icône **?** au niveau de toutes les tâches utilisateur d'un processus, il faut préciser la référence de cette aide en ligne dans le champ **Référence de l'aide en ligne** prévu à cet effet. Ce champ se trouve dans l'onglet **Propriétés**, rubrique **Autres propriétés**.



Panneau à afficher lors de l'ouverture d'une tâche sur un petit écran

Lorsqu'un écran est trop étroit pour afficher simultanément le formulaire de saisie et le panneau d'affichage des documents (tablette en mode portrait par exemple), l'option **Sur petit écran, ouvrir la tâche sur** permet de choisir l'élément à afficher en priorité (le formulaire ou le panneau de documents) lors de l'ouverture de la tâche.

Cette option se trouve dans l'onglet **Affichage** d'une tâche utilisateur ou d'une tâche de démarrage.

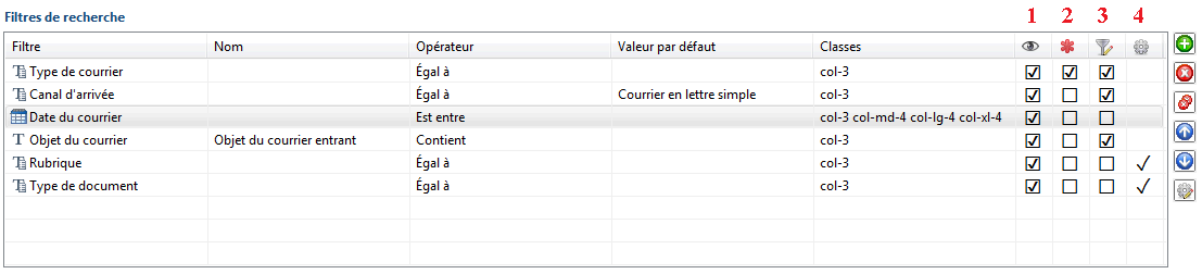


> ***Même si on choisit d'ouvrir le volet d'affichage des documents, le formulaire est toujours chargé en arrière-plan. Seuls le comportement de certains effets visuels personnalisés appelés lors du chargement du formulaire peuvent être perturbés (comme le positionnement et la focalisation sur élément en particulier).***

Configuration du panneau de recherche multicritères

Chaque processus peut inclure un panneau de recherche multicritères prédéfini, permettant aux utilisateurs de sélectionner les tâches correspondant à leurs critères de recherche.

Cette configuration s'effectue dans l'onglet **Affichage et recherches,** au niveau de la rubrique **Filtres de recherche.**



La méthode pour spécifier les filtres à afficher sur le panneau de recherche suit le même principe que la configuration des filtres dans l'extension **Sélecteur de documents**.

Les quatre dernières colonnes représentent :

1 : filtre visible ou non dans l'interface

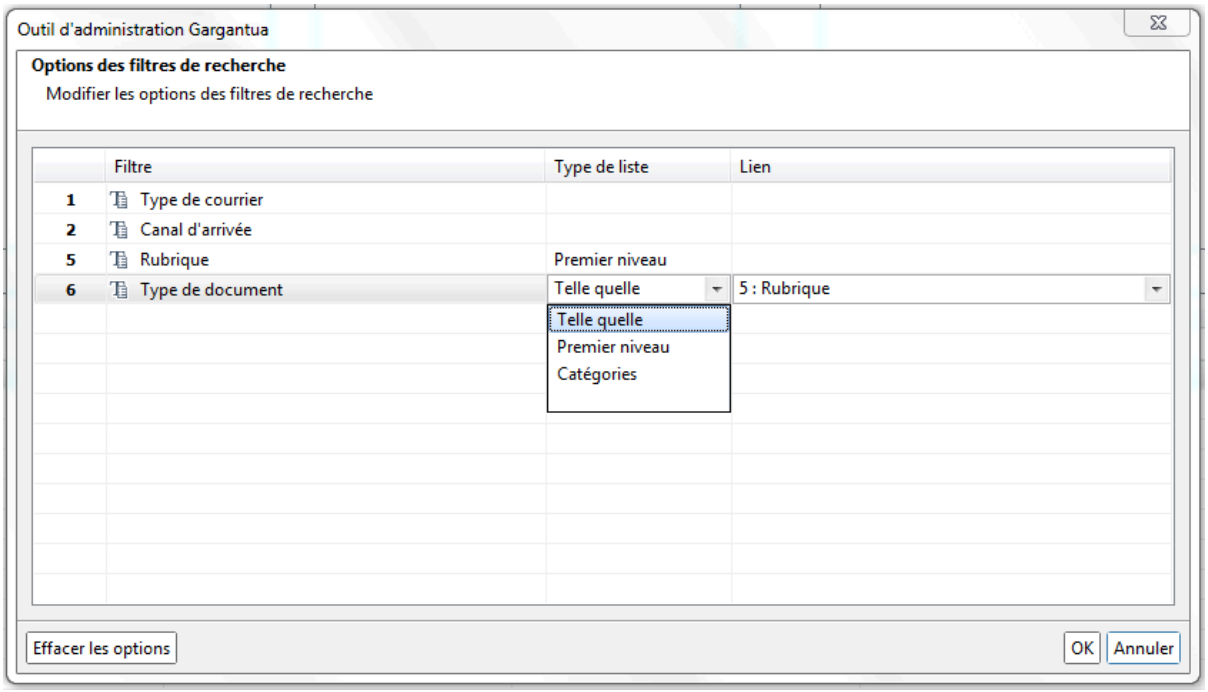
2 : filtre obligatoire ou non

3 : autorise ou non le changement de l'opérateur dans l'interface utilisateur. Par défaut, le changement d'opérateur n'est pas autorisé.

Pour les filtres utilisant des opérateurs d'intervalle (est entre et n'est pas entre), il est fortement recommandé de prévoir des classes suffisamment larges (au moins col-4/col-md-4,col-lg-4, col-xl-4), car dans ce cas deux champs de saisie sont affichés.

*Même si l'opérateur initial n'est pas un opérateur d'intervalle (**est entre** ou **« n'est pas entre »**), si la propriété **« Autoriser le changement d'opérateur »** est cochée, les classes de critères doivent également être choisies en en tenant compte.*

4 : options des filtres - pour le moment les options sont spécifiques aux champs d'indexation de type liste (identiques aux options déjà présentes dans les modèles de recherche multicritères).



Les champs d'indexation de type zone de texte peuvent être utilisés comme filtres si la base est configurée en texte intégral.

Pour les filtres représentant des champs d'indexation de type alphanumérique dont la propriété 'recherche exacte' est cochée, les opérateurs disponibles sont restreints à **Egal à** et **Est vide** (comme c'est le cas dans le module de recherche multicritères).

Gestion des modèles de document

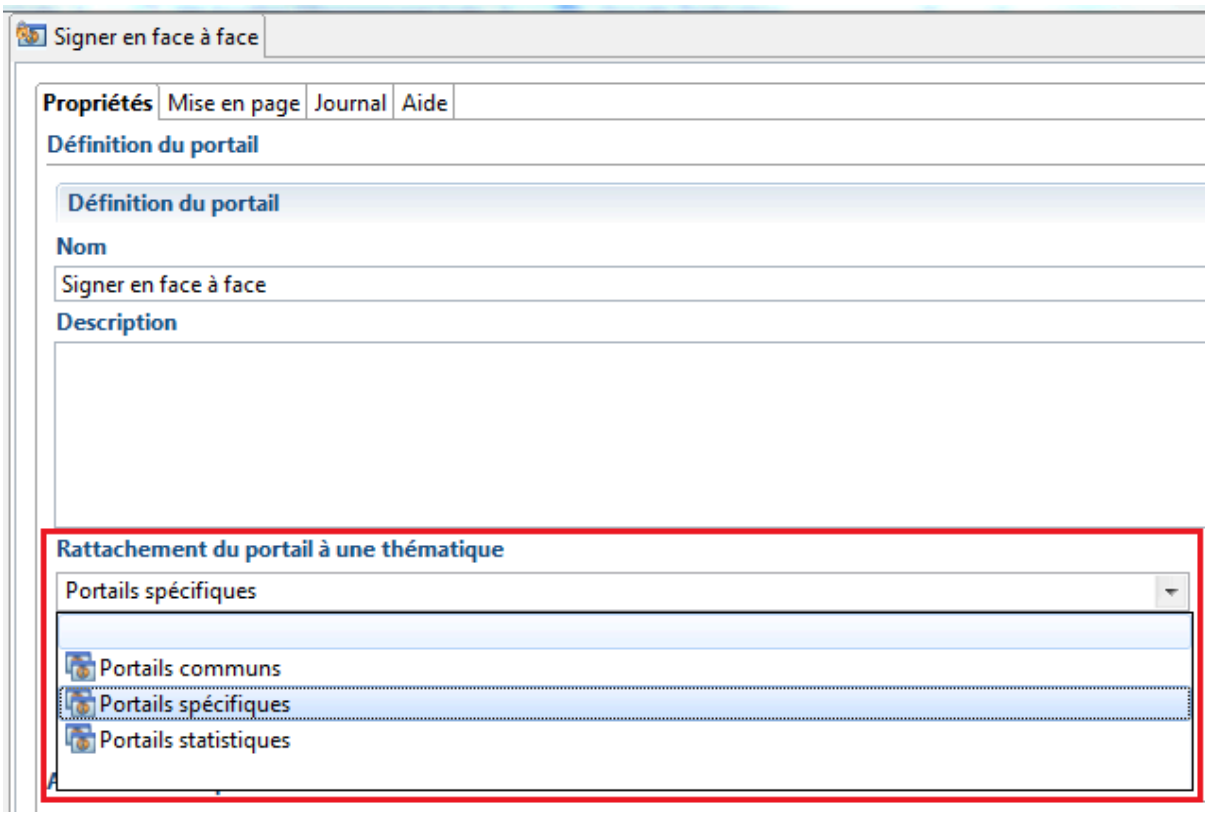
Formats de documents pouvant être ouverts dans leur application d'origine

Depuis la version 8.3, il est possible d'ouvrir les documents **PowerPoint** dans leur application d'origine, au même titre que les documents **Word** et **Excel**.

Gestion des portails

Rattachement d'un portail à une thématique

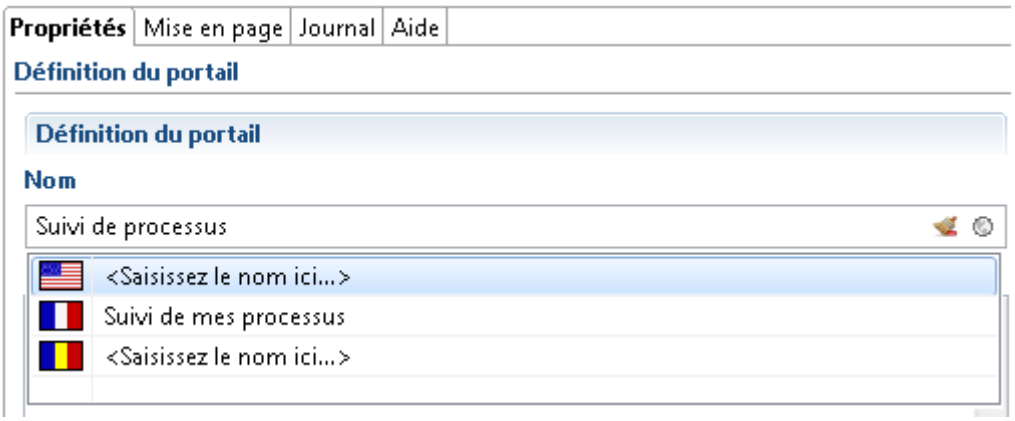
Un portail peut être associé à une thématique définie au préalable dans la section **Modules Portails / Regroupement thématique des portails** dans un menu déroulant. Suite à ce rattachement, un menu déroulant contenant les différentes thématiques est affiché lorsqu'un utilisateur accède au module **Portails**.



Personnalisation des noms des portails

Les noms des portails peuvent être traduits dans toutes les langues paramétrées au niveau de la configuration

du serveur.



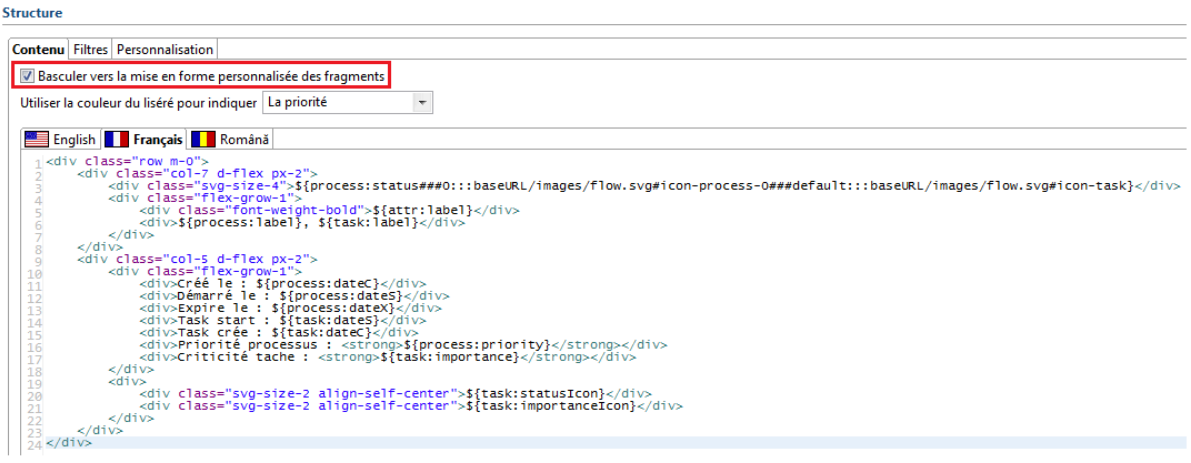
CONFIGURATION DES MODULES

MODULE TÂCHES

UTILISATION DE BOOTSTRAP POUR LA MISE EN FORME DES FRAGMENTS

L'outil de mise en forme des fragments permet de passer d'une mise en forme Tableau à une mise en forme personnalisée.

Lorsqu'on bascule en mode **Mise en forme personnalisée**, on peut écrire du code HTML et utiliser des classes Bootstrap pour rendre l'affichage réactif.



Lorsque le mode **Mise en forme personnalisée** est activé, c'est cette configuration qui est utilisée, même en présence de paramétrage de la **mise en forme Tableau**. Cependant, la configuration **mode Tableau** est conservée, permettant ainsi un retour facile à cette mise en forme si nécessaire.

Important : Si vous revenez à la mise en forme Tableau, vous perdez toute configuration réalisée dans la mise en forme personnalisée.

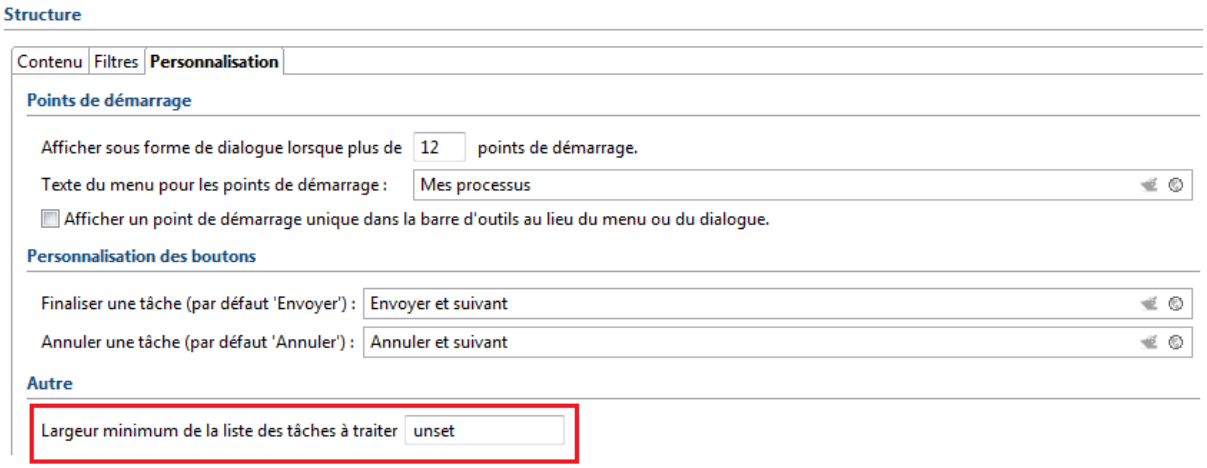
Le changement de la mise en forme du fragment s'applique dans tous les fragments : dans web pour le module de taches, dans l'outil de conception de processus, dans les fragments des formulaires dans les applications Administrateur et Web.

Configuration de la largeur minimale de la liste des tâches à traiter

Lorsque le mode **Tableau** est retenu pour mettre en forme les fragments, il est nécessaire d'indiquer la largeur minimale à réserver sur l'écran pour l'affichage de la liste des tâches à traiter. La valeur par défaut est 900px.

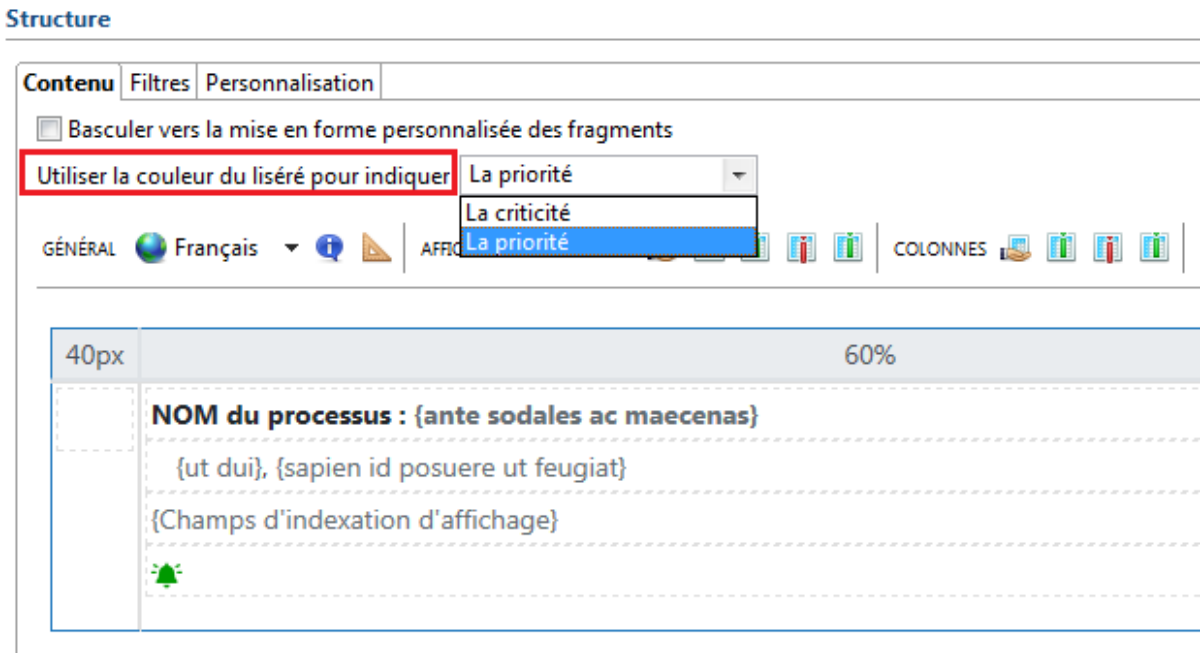
En revanche, lorsque le mode **Mise en forme personnalisée** est retenu, ce paramètre peut soit :

- Rester vide, pour assurer la compatibilité avec les anciens fragments. Dans ce cas, c'est la largeur minimale de 900px qui s'applique.
- Être défini avec une valeur CSS valide telle que %, px, rem, ...
- Être désactivé, en positionnant la valeur **unset**.



Représentation de la priorité ou de la criticité par la couleur du liséré d'une tâche

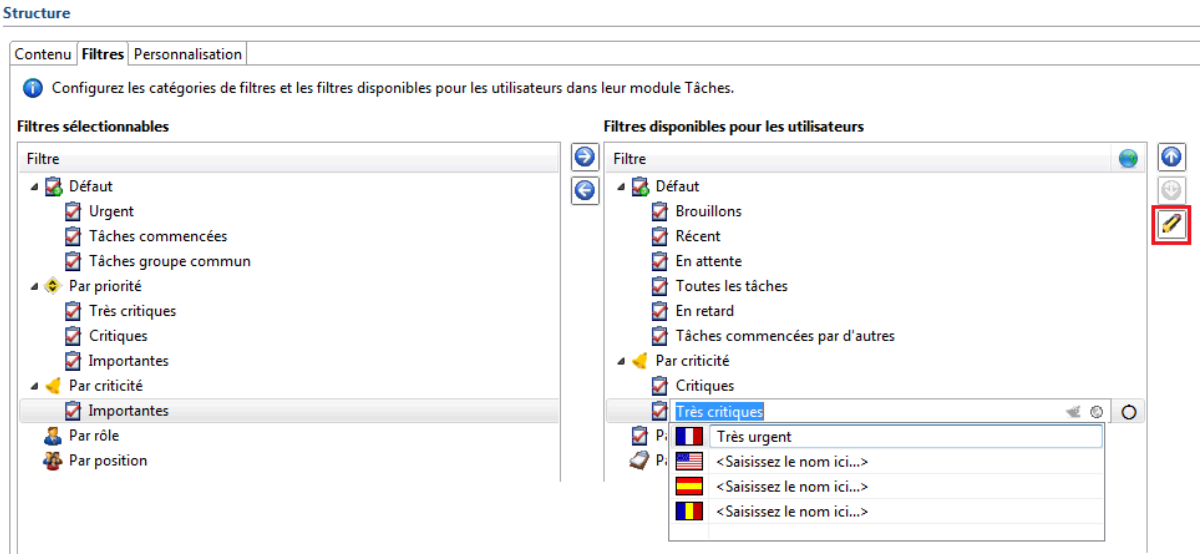
Par défaut, la couleur du liséré indique la priorité du processus. Cependant, vous pouvez configurer cette couleur pour qu'elle reflète la priorité de la criticité d'une tâche à traiter, en ajustant le paramètre **Couleur du liséré**.



*Dans le module de **Supervision des processus**, la couleur du liséré indique toujours la priorité du processus (cette couleur ne peut en aucun cas représenter la criticité d'une tâche car c'est la liste des processus qui est affichée pour leur supervision).*

Personnalisation des libellés des catégories de filtre et des filtres

Dans le panneau des filtres, les libellés peuvent être personnalisés dans toutes les langues configurées au niveau de la base documentaire. Cette personnalisation concerne aussi bien les catégories des filtres que les filtres eux-mêmes.



Les catégories de filtres et les filtres avec des libellés personnalisés sont signalés par une icône de cercle plein. Quand le cercle est vide, cela signifie qu'au moins une traduction de libellé est renseignée. Quand aucune personnalisation n'est renseignée, aucun cercle n'est présent.

L'utilisation de l'icône **Modifier** est obligatoire.

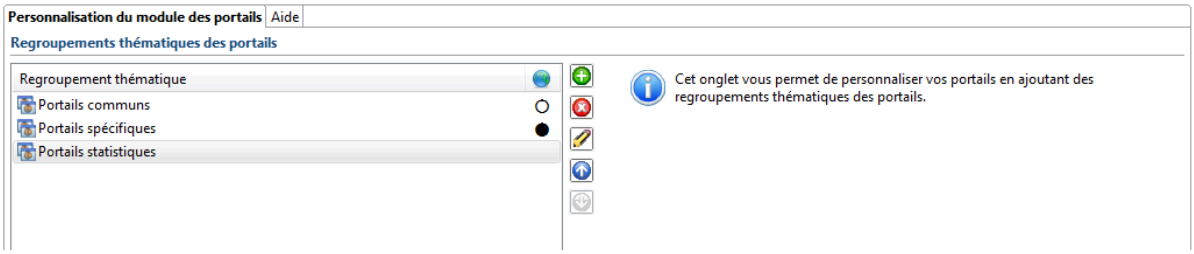
> ***Pour personnaliser le libellé des filtres, il est nécessaire d'utiliser l'icône Modifier, car la fonction de double-clic est déjà utilisée pour l'ajout et la suppression de catégories et de filtres.***

Module Portails

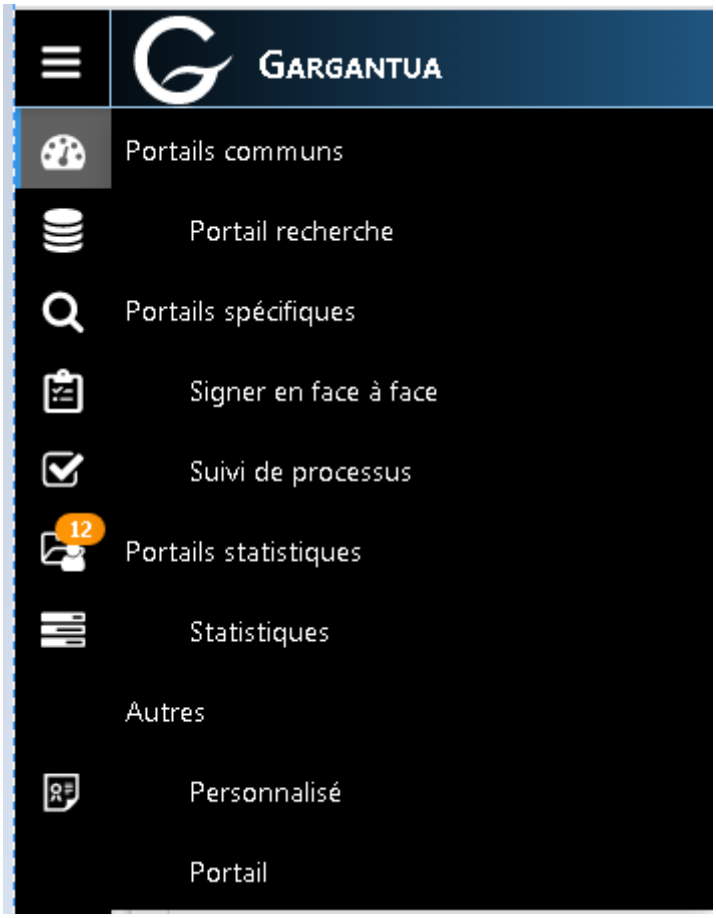
Regroupement thématique des portails dans un menu déroulant

Pour les utilisateurs ayant accès à une multitude de portails, tels que divers portails de statistiques et de recherche par exemple, cette fonctionnalité offre la possibilité de les regrouper par thèmes personnalisés. Ces groupes thématiques sont ensuite accessibles via un menu déroulant dans le module Portails, facilitant la navigation de l'utilisateur.

- Pour **ajouter** une thématique, utilisez l'icône +.
- Pour la **supprimer**, utilisez l'icône -.
- Pour **modifier leur ordre d'affichage** dans le menu déroulant, utilisez les icônes **Flèche vers le haut** et **Flèche vers le bas**.
- Pour **traduire les libellés** des différentes thématiques dans toutes les langues disponibles dans la base documentaire, utilisez l'icône **Globe** située à côté de chaque thématique.



Une fois les thématiques créées, elles peuvent être utilisées dans la rubrique **Gestion des portails** pour configurer le rattachement d'un portail à une thématique donnée.



MODÈLES DE NOTIFICATION

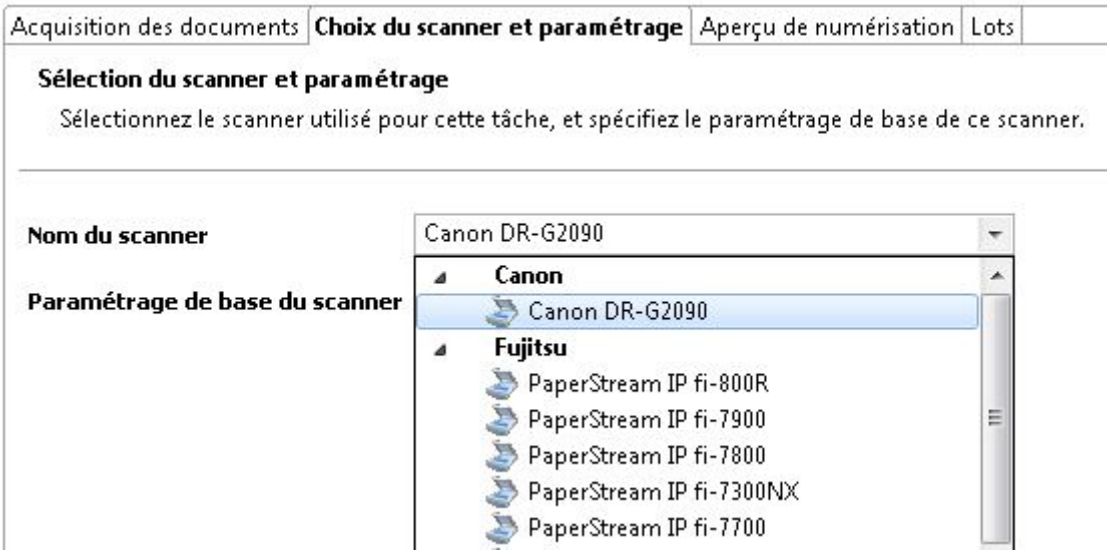
ENVOI DU CODE DE SÉCURITÉ POUR L'AUTHENTIFICATION À DEUX FACTEURS

Le modèle de mail **Code d'authentification à 2 facteurs**, destiné à l'envoi du code de sécurité nécessaire à la double authentification, est accessible dans la section **Modèles d'utilisateur**.

Tâches IntelliScan

Compatibilité avec le driver du scanner Canon DR-G2090

Depuis la version 8.3, le scanner Canon DR-G2090 peut être utilisé pour les tâches IntelliScan.



APIs scripting serveur

Modification du champ d'indexation date/heure

La valeur d'un champ d'indexation enregistrée par script était jusqu'à présent un objet date/heure complet, comprenant les secondes et les millisecondes.

Mais l'interface utilisateur ne permettait ni d'afficher les secondes ni les millisecondes.

Désormais, lors de l'enregistrement des valeurs des champs d'indexation date/heure, les secondes et les millisecondes seront supprimées de la valeur.

Pour mettre à jour les données existantes, si nécessaire, il faut exécuter une requête comme celle-ci (remplacez `siatel_g00003` par le bon identifiant de la base Gargantua et le `attr_0000000200` par le bon identifiant d'attribut de type date/heure) :

```
update siatel_g00003.attr_docs_f_00011 set attr_0000000200 =
((attr_0000000200/(60*1000))*60*1000) where attr_0000000200 is not null;
```

Cette opération est nécessaire dans le cas ou des statistiques sont définies sur la base et l'attribut date/heure est renseigné automatiquement par des scripts.

Amélioration de l'énumération des rôles et des groupes d'un utilisateur

Les commandes `CMD_ENUM_ROLES` et `CMD_ENUM_GROUPS` peuvent maintenant être utilisées en renseignant un **userId** directement dans le paramètre **dtoIn**.

Méthode Ged.deleteFolder

La fonction **Ged.deleteFolder** permet la suppression d'un dossier, ce dossier est donc toujours déplacé vers la corbeille de la base.

Il n'y pas de confirmation avant suppression.

Ce comportement sécurise le fonctionnement en donnant la possibilité de restauration du dossier dans le cas d'une suppression de ce dossier par accident.

Intégration d'Apache OpenNLP comme moteur de détection de la langue

> ***Le modèle Apache OpenNLP utilisé détecte 103 langues selon la norme ISO 693-3 et fonctionne bien avec des textes plus longs contenant au moins 2 phrases ou plus dans la même langue.***

Le modèle de langages utilisé est téléchargé depuis : https://opennlp.apache.org/models.html , c'est le fichier « langdetect-183.bin ». Dans le document README lié a ce fichier on peut lire « **The model can detect 103 languages in ISO 639-3 standard** » et dedans il y a aussi la liste avec les langages connus : https://downloads.apache.org/opennlp/models/langdetect/1.8.3/README.txt

```
var resultatEnFormatJSON = CustomExtensions.detectLanguage(idOrFile);
```

'idOrFile' peut être :

- l'identifiant d'un document
- l'identifiant d'un fichier
- le chemin d'un fichier

Carnet d'adresses v2

Voici une liste non exhaustive de fonctions et méthodes pour la version du carnet d'adresses apportée par la V8.3 de Gargantua. On y trouve des fonctions « CRUD ». Les noms de fonctions et méthodes sont assez explicites (enumerateContacts, createPerson, setSurname,...) :

```
//
// Fonctions génériques
//
////////////////////////////////////

function enumerateContacts() {
    var contacts = new java.util.ArrayList(0);
    var criteria = DTOFactory.newDTO();
    var json = '{ "contactList" : [ ';
    try {
        AddressBook2.enumerateContacts(criteria, contacts);
        if (contacts != null && !contacts.isEmpty()) {
            for (var i = 0; i < contacts.size(); i++) {
                var contact = contacts.get(i);
                var contactDTO = contact.toDTO();
                var contactId = contactDTO.get("id");
                var contactName = contactDTO.get("surname") != null ? contactDTO
                        .get("surname")
                        : contactDTO.get("name");
                var contactType = contactDTO.get("surname") != null ? "person"
                        : "organisation";
                if (i > 0) {
                    json = json + ', ';
                }
                json = json + '{ "contactId" : "' + contactId
                        + '", "contactName" : "' + contactName
                        + '", "contactType" : "' + contactType + '"}';
            }
        }
    } catch (e) {
    }
    json = json + ' ] }';
    return json;
}

////////////////////////////////////
//
// Person
//
////////////////////////////////////

function createPerson() {
    var CONTACT_TYPE = "person";
    var PERSON_TITLE_MR = 1;
    var COUNTRY_ISO_3 = "FRA";

    var person = AddressBook2.newContact(CONTACT_TYPE);
    person.setDateC(new java.util.Date());

    person.setSurname("Moire");
    person.setForename("Emmanuel");
    person.setAlias("emmi");
    person.setEmail("emmanuel.moire@gmail.com");

    person.setTitle(PERSON_TITLE_MR);
    person.setTitleProf("Chanteur");

    person.setUrl1("https://www.test.com");
    person.setUrl2("https://www.linkedin.test.com");
    person.setNote("Emmanuel Moire - un chanteur français");

    person.setVip(false);
    person.setMoved(false);
    person.setPnta(false);
```

```
    person.setDead(false);
    person.setElected(false);
    person.setUnionRep(false);
    person.setPublicPerson(true);

    var homeAddress = AddressBook2.newAddress();
    homeAddress.setStreet1("Rue de Paris");
    homeAddress.setStreet2("No 53");
    homeAddress.setStreet3("");

    homeAddress.setPostcode("91120");
    homeAddress.setCity("Palaiseau");
    homeAddress.setRegion("Île-de-France");
    homeAddress.setCountry(COUNTRY_ISO_3);

    person.setAddress(homeAddress);

    AddressBook2.createContact(person);

    return "Contact id + surname : " + person.getId() + " + "
            + person.getSurname();
}

function createPerson2() {
    var PERSON_TITLE_MR = 1;
    var COUNTRY_ISO_3 = "ROU";

    var person = AddressBook2.newPerson();
    person.setDateC(new java.util.Date());

    person.setSurname("Timberlake");
    person.setForename("Justin");
    person.setAlias("just");
    person.setEmail("justin.timberlake@gmail.com");

    person.setTitle(PERSON_TITLE_MR);
    person.setTitleProf("Songwriter");

    person.setUrl1("https://www.test.com");
    person.setUrl2("https://www.linkedin.test.com");
    person.setNote("American singer-songwriter and record producer");

    person.setVip(false);
    person.setMoved(false);
    person.setPnta(false);
    person.setDead(false);
    person.setElected(false);
    person.setUnionRep(false);
    person.setPublicPerson(true);

    var street1 = "Tei";
    var street2 = "No 22";
    var postcode = "032171";
    var city = "Bucuresti";
    var region = "Sector 3";
    var country = COUNTRY_ISO_3;

    var homeAddress = AddressBook2.newAddress(DTOFactory.newDTO("street1",
            street1, "street2", street2, "postcode", postcode, "city", city,
            "region", region, "country", country));
    person.setAddress(homeAddress);

    AddressBook2.createPerson(person);
```

```
    return "Contact id + surname : " + person.getId() + " + "
            + person.getSurname();
}

function createPerson3() {
    var CONTACT_TYPE = "person";
    var COUNTRY_ISO_3 = "ROU";
    var PERSON_TITLE_MR = 1;

    var surname = "Roslansky";
    var forename = "Ryan";
    var alias = "rr";
    var email = "ryan.roslansky@gmail.com";
    var title = PERSON_TITLE_MR;
    var titleProf = "CEO";
    var url1 = "https://www.ryanroslansky.com";
    var url2 = "https://www.linkedin.com/in/ryanroslansky";
    var note = "Note : contact";

    var vip = false;
    var moved = true;
    var pnta = false;
    var dead = false;
    var elected = false;
    var unionRep = false;
    var publicPerson = true;

    var street1 = "Strada Zambilelor";
    var street2 = "Nr 23";
    var street3 = "Bloc 12, Ap 20";
    var postcode = "032174";
    var city = "Bucuresti";
    var region = "Sector 2";
    var country = COUNTRY_ISO_3;

    var phoneList = new java.util.ArrayList(0);
    phoneList.add(DTOFactory.newDTO("number", "+40767675645", "type", "1"));
    phoneList.add(DTOFactory.newDTO("number", "+40250303332", "type", "2"));

    var person = AddressBook2.newContact(DTOFactory.newDTO("contactType",
            CONTACT_TYPE, "surname", surname, "forename", forename, "alias",
            alias, "email", email, "title", title, "titleProf", titleProf,
            "url1", url1, "url2", url2, "note", note, "vip", vip, "moved",
            moved, "pnta", pnta, "dead", dead, "elected", elected, "unionRep",
            unionRep, "publicPerson", publicPerson, "street1", street1,
            "street2", street2, "street3", street3, "postcode", postcode,
            "city", city, "region", region, "country", country, "phones",
            phoneList));

    AddressBook2.createContact(person);

    return "Contact id + surname : " + person.getId() + " + "
            + person.getSurname();
}

function readPerson() {
    var CONTACT_TYPE = "person";

    var contactId = 73518;
    var person = AddressBook2.readContact(contactId, CONTACT_TYPE);

    return "Contact surname + forename : " + person.getSurname() + " + "
            + person.getForename();
}
```

```
function readPerson2() {
    var contactId = 73512;
    var person = AddressBook2.readPerson(contactId);

    return "Contact surname + alias : " + person.getSurname() + " + "
            + person.getAlias();
}

function updatePerson() {
    var CONTACT_TYPE = "person";
    var COUNTRY_ISO_3 = "ROU";

    var contactId = 73518;
    var name = "";

    var street1 = "Tei";
    var street2 = "No 22";
    var street3 = "No 24";

    var postcode = "032171";
    var city = "Bucuresti";
    var region = "Sector 3";
    var country = COUNTRY_ISO_3;

    try {
        var person = AddressBook2.readContact(contactId, CONTACT_TYPE);
        person.setSurname("Anghel");
        person.setForename("Marcel");
        person.setAlias("AM");
        person.setEmail("anghel.marcel@gmail.com");

        var address = person.getAddress();
        if (address != null) {
            address.setStreet1(street1);
            address.setStreet2(street2);
            address.setStreet3(street3);
            address.setCountry(country);
        } else {
            address = AddressBook2.newAddress(DTOFactory.newDTO("street1",
                    street1, "street2", street2, "postcode", postcode, "city",
                    city, "region", region, "country", country));

            person.setAddress(address);
        }

        AddressBook2.updateContact(person);

        name += person.getSurname() + " " + person.getForename() + " "
                + person.getAlias();

    } catch (e) {

    }

    return "Person : " + name;
}

function updatePerson2() {
    var contactId = 73518;
    var name = "";

    try {
        var person = AddressBook2.readPerson(contactId);
```

```
        person.setSurname("Anghel_v2");
        person.setForename("Marcel_v2");
        person.setAlias("AM_v2");
        person.setEmail("anghel.marcel_v2@gmail.com");

        AddressBook2.updatePerson(person);

        name = person.getSurname() + " " + person.getForename() + " "
                + person.getAlias();
    } catch (e) {

    }

    return "Person : " + name;
}

function updatePerson3() {
    var contactId = 73518;
    var name = "";

    try {
        var person = AddressBook2.readPerson(contactId);
        person.setPublicPerson(true);
        person.setAlias("smiley");

        var phoneList = new java.util.ArrayList(0);
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40750665523", "type", "1")));
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40250405090", "type", "2")));

        person.setPhones(new java.util.HashSet(phoneList));

        AddressBook2.updatePerson(person);

        name = person.getSurname() + " " + person.getForename() + " "
                + person.getAlias();
    } catch (e) {

    }

    return "Person : " + name;
}

function idOfPerson() {
    var surname = "Roslansky";
    var contactType = "person";
    var id = AddressBook2.idOfContact(surname, contactType);

    return "Person id + surname : " + id + " " + surname;
}

function idOfPerson2() {
    var surname = "Roslansky";
    var id = AddressBook2.idOfPerson(surname);

    return "Person id + surname : " + id + " " + surname;
}

function addPersonRelationalLink() {
    var personId = 75895;
    var personRlinkId = 71160;

    var PERSON_RLINK_BROTHER = 5;
```

```
    AddressBook2.addPersonRelationalLink(personId, personRlinkId,
            PERSON_RLINK_BROTHER);
}

function updatePersonRelationalLink() {
    var personId = 75895;
    var personRlinkId = 71160;

    var PERSON_RLINK_HALF_BROTHER = 8;

    AddressBook2.updatePersonRelationalLink(personId, personRlinkId,
            PERSON_RLINK_HALF_BROTHER);
}

function removePersonRelationalLink() {
    var personId = 75895;
    var personRlinkId = 71160;

    AddressBook2.removePersonRelationalLink(personId, personRlinkId);
}

function enumeratePersonRelationalLinks() {
    var personId = 75895;
    var PERSON_RLINK_BROTHER = 5;

    var rlinks = new java.util.ArrayList(0);
    var criteria = DTOFactory.newDTO("personId", personId, "rlinkType",
            PERSON_RLINK_BROTHER);

    AddressBook2.enumeratePersonRelationalLinks(criteria, rlinks);

    var json = '{ "rlinks" : [ ';

    if (rlinks != null && !rlinks.isEmpty()) {

        for (var i = 0; i < rlinks.size(); i++) {

            var rlink = rlinks.get(i);
            var person = rlink.xget("person");
            var rlinkType = rlink.getInteger("rlinkType");

            if (i > 0) {
                json = json + ', ';
            }

            json = json + '{ "personId" : "' + person.getId()
                    + '",  "rlinkType" : "' + rlinkType + '"}';
        }
    }

    json = json + ' ] }';
    return json;

}

function deletePerson() {
    var id = 75835;
    try {
        AddressBook2.deletePerson(id);

    } catch (e) {

    }
```

```
}

function enumeratePersons() {

    var persons = new java.util.ArrayList(0);
    var criteria = DTOFactory.newDTO();
    var json = '{ "persons" : [ ';
    try {
        AddressBook2.enumeratePersons(criteria, persons);
        if (persons != null && !persons.isEmpty()) {
            for (var i = 0; i < persons.size(); i++) {
                var person = persons.get(i);
                var name = person.getSurname() + " " + person.getForename();
                var alias = person.getAlias() != null ? person.getAlias() : "";
                if (i > 0) {
                    json = json + ', ';
                }
                json = json + '{ "id" : "' + person.getId() + '", "name" : "'
                        + name + '", "alias" : "' + alias + '"}';

            }
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}

function enumeratePersons2() {

    var persons = new java.util.ArrayList(0);

    var criteria = DTOFactory.newDTO();
    criteria.put("email", "siatel");
    criteria.put("city", "Bucuresti");

    var json = '{ "persons" : [ ';

    try {

        AddressBook2.enumeratePersons(criteria, persons);

        if (persons != null && !persons.isEmpty()) {

            for (var i = 0; i < persons.size(); i++) {

                var person = persons.get(i);
                var name = person.getSurname() + " " + person.getForename();
                var alias = person.getAlias() != null ? person.getAlias() : "";

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "id" : "' + person.getId() + '", "name" : "'
                        + name + '", "alias" : "' + alias + '"}';

            }
        }
    } catch (e) {
```

```
    }

    json = json + ' ] }';

    return json;
}

////////////////////////////////////
//
// Organisation
//
////////////////////////////////////

function createOrganisation() {
    var CONTACT_TYPE = "organisation";
    var COUNTRY_ISO_3 = "FRA";

    var ORGANISATION_TYPE_ASSOCIATION = 1;

    var organisation = AddressBook2.newContact(CONTACT_TYPE);
    organisation.setDateC(new java.util.Date());

    organisation.setName("Vodafone");
    organisation.setAcronym("vdf");
    organisation.setEmail("contact.vodafone@gmail.com");
    organisation.setTin("909099");

    organisation.setType(ORGANISATION_TYPE_ASSOCIATION);

    organisation.setUrl1("https://www.test.com");
    organisation.setUrl2("https://www.linkedin.test.com");
    organisation.setNote("Note Vodafone");

    organisation.setVip(false);
    organisation.setMoved(false);
    organisation.setPnta(true);
    organisation.setClosed(false);

    var address = AddressBook2.newAddress();
    address.setStreet1("Rue de Paris");
    address.setStreet2("No 53");
    address.setStreet3("");

    address.setPostcode("91120");
    address.setCity("Palaiseau");
    address.setRegion("Île-de-France");
    address.setCountry(COUNTRY_ISO_3);

    organisation.setAddress(address);

    AddressBook2.createContact(organisation);

    return "Contact id + name : " + organisation.getId() + " + "
            + organisation.getName();
}

function createOrganisation2() {
    var COUNTRY_ISO_3 = "ROU";

    var organisation = AddressBook2.newOrganisation();
    organisation.setDateC(new java.util.Date());

    organisation.setName("Farmacia Tei");
    organisation.setEmail("contact.farmaciatei@gmail.com");
```

```
    organisation.setUrl1("https://www.test.com");
    organisation.setUrl2("https://www.linkedin.test.com");
    organisation.setNote("Note : Farmacia Tei");

    organisation.setVip(false);
    organisation.setMoved(false);
    organisation.setPnta(false);
    organisation.setClosed(true);

    var street1 = "Tei";
    var street2 = "No 22";
    var postcode = "032171";
    var city = "Bucuresti";
    var region = "Sector 3";
    var country = COUNTRY_ISO_3;

    var address = AddressBook2.newAddress(DTOFactory.newDTO("street1", street1,
            "street2", street2, "postcode", postcode, "city", city, "region",
            region, "country", country));
    organisation.setAddress(address);

    AddressBook2.createOrganisation(organisation);

    return "Contact id + name : " + organisation.getId() + " + "
            + organisation.getName();
}

function createOrganisation3() {
    var CONTACT_TYPE = "organisation";
    var COUNTRY_ISO_3 = "ROU";
    var ORGANISATION_TYPE_COMPANY = 2;

    var name = "IT Teams";
    var acronym = "itt";
    var tin = "009988";
    var service = "Software";

    var email = "contact.itteams@gmail.com";
    var orgType = ORGANISATION_TYPE_COMPANY;
    var url1 = "https://www.test.com";
    var url2 = "https://www.linkedin.com/in/itteams";
    var note = "Note : it teams";

    var vip = true;
    var moved = true;
    var pnta = false;
    var closed = false;

    var street1 = "Strada Zambilelor";
    var street2 = "Nr 23";
    var street3 = "Bloc 12, Ap 20";
    var postcode = "032174";
    var city = "Bucuresti";
    var region = "Sector 2";
    var country = COUNTRY_ISO_3;

    var phoneList = new java.util.ArrayList(0);
    phoneList.add(DTOFactory.newDTO("number", "+40733675645", "type", "1"));
    phoneList.add(DTOFactory.newDTO("number", "+40250303112", "type", "2"));

    var organisation = AddressBook2.newContact(DTOFactory.newDTO("contactType",
            CONTACT_TYPE, "name", name, "acronym", acronym, "email", email,
            "orgType", orgType, "tin", tin, "service", service, "url1", url1,
            "url2", url2, "note", note, "vip", vip, "moved", moved, "pnta",
```

```
            pnta, "closed", closed, "street1", street1, "street2", street2,
            "street3", street3, "postcode", postcode, "city", city, "region",
            region, "country", country, "phones", phoneList));

    AddressBook2.createContact(organisation);

    return "Contact id + name : " + organisation.getId() + " + "
            + organisation.getName();
}

function readOrganisation() {
    var CONTACT_TYPE = "organisation";

    var contactId = 75660;
    var organisation = AddressBook2.readContact(contactId, CONTACT_TYPE);

    return "Contact name + acronym : " + organisation.getName() + " + "
            + organisation.getAcronym();
}

function readOrganisation2() {
    var contactId = 75627;
    var organisation = AddressBook2.readOrganisation(contactId);

    return "Contact name + acronym : " + organisation.getName() + " + "
            + organisation.getAcronym();
}

function updateOrganisation() {
    var CONTACT_TYPE = "organisation";
    var COUNTRY_ISO_3 = "ROU";

    var contactId = 75627;
    var name = "";

    var street1 = "Tei";
    var street2 = "No 22";
    var street3 = "No 24";

    var postcode = "032171";
    var city = "Bucuresti";
    var region = "Sector 3";
    var country = COUNTRY_ISO_3;

    try {
        var organisation = AddressBook2.readContact(contactId, CONTACT_TYPE);
        organisation.setName("Vodafone RO");
        organisation.setEmail("contact.vodafone.ro@gmail.com");

        var address = organisation.getAddress();
        if (address != null) {
            address.setStreet1(street1);
            address.setStreet2(street2);
            address.setStreet3(street3);
            address.setCountry(country);
        } else {
            address = AddressBook2.newAddress(DTOFactory.newDTO("street1",
                    street1, "street2", street2, "postcode", postcode, "city",
                    city, "region", region, "country", country));

            organisation.setAddress(address);
        }

        AddressBook2.updateContact(organisation);
```

```
        name += organisation.getName() + " " + organisation.getAcronym();

    } catch (e) {

    }

    return "Organisation : " + name;
}

function updateOrganisation2() {
    var contactId = 71037;
    var name = "";

    try {
        var organisation = AddressBook2.readOrganisation(contactId);
        organisation.setName("Dr Max RO");
        organisation.setAcronym("max");
        organisation.setEmail("contact.drmax@gmail.com");

        AddressBook2.updateOrganisation(organisation);

        name = organisation.getName() + " - " + organisation.getAcronym();
    } catch (e) {

    }

    return "Organisation : " + name;
}

function updateOrganisation3() {
    var contactId = 71220;
    var name = "";

    try {
        var organisation = AddressBook2.readOrganisation(contactId);
        organisation.setPnta(true);
        organisation.setAcronym("ing");

        var phoneList = new java.util.ArrayList(0);
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40750665523", "type", "1")));
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40250405090", "type", "2")));

        organisation.setPhones(new java.util.HashSet(phoneList));

        AddressBook2.updateOrganisation(organisation);

        name = organisation.getName() + " " + organisation.getAcronym();
    } catch (e) {

    }

    return "Organisation : " + name;
}

function idOfOrganisation() {
    var name = "Vodafone RO";
    var contactType = "organisation";
    var id = AddressBook2.idOfContact(name, contactType);

    return "Organisation id + name : " + id + " " + name;
}
```

```
function idOfOrganisation2() {
    var name = "ING ROMANIA";
    var id = AddressBook2.idOfOrganisation(name);

    return "Organisation id + name : " + id + " " + name;
}

function deleteOrganisation() {
    var id = 75875;

    try {
        AddressBook2.deleteOrganisation(id);

    } catch (e) {

    }
}

function enumerateOrganisations() {

    var organisations = new java.util.ArrayList(0);
    var criteria = DTOFactory.newDTO();

    var json = '{ "organisations" : [ ';

    try {

        AddressBook2.enumerateOrganisations(criteria, organisations);

        if (organisations != null && !organisations.isEmpty()) {

            for (var i = 0; i < organisations.size(); i++) {

                var organisation = organisations.get(i);
                var name = organisation.getName();
                var acronym = organisation.getAcronym != null ? organisation
                        .getAcronym() : "";

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "id" : "' + organisation.getId()
                        + '", "name" : "' + name + '", "acronym" : "' + acronym
                        + '"}';

            }
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}

function enumerateOrganisations2() {

    var organisations = new java.util.ArrayList(0);

    var criteria = DTOFactory.newDTO();
    criteria.put("orgName", "Farmacia");
```

```
    criteria.put("onlyClosed", true);

    var json = '{ "organisations" : [ ';

    try {

        AddressBook2.enumerateOrganisations(criteria, organisations);

        if (organisations != null && !organisations.isEmpty()) {

            for (var i = 0; i < organisations.size(); i++) {

                var organisation = organisations.get(i);
                var name = organisation.getName();
                var acronym = organisation.getAcronym != null ? organisation
                        .getAcronym() : "";

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "id" : "' + organisation.getId()
                        + '", "name" : "' + name + '", "acronym" : "' + acronym
                        + '"}';

            }
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}

////////////////////////////////////
//
// Assignments
//
////////////////////////////////////

function readAssignment() {
    var personId = 75719;
    var organisationId = 75660;

    var assignment = AddressBook2.readAssignment(personId, organisationId);
    var email = assignment.getEmail();

    var departments = assignment.getDepartments() != null ? (new java.util.ArrayList(
            assignment.getDepartments()))
            : null;
    var department = departments != null ? departments.get(0) : null;
    var deptName = department != null ? department.getDepartment() : "";
    var position = department != null ? department.getPosition() : "";

    var mobile = "";

    var phones = assignment.getPhones() != null ? (new java.util.ArrayList(
            assignment.getPhones())) : null;

    if (phones != null) {
        for (var i = 0; i < phones.size(); i++) {
            var phone = phones.get(i);
```

```
            if (phone.getType() == 1) {
                mobile = phone.getNumber();
            }
        }
    }

    return " Assignment : " + email + " " + deptName + " " + position + " "
            + mobile;
}

function updateAssignment() {
    var personId = 75895;
    var organisationId = 71256;

    var assignment = AddressBook2.readAssignment(personId, organisationId);

    try {

        assignment.setEmail("contact.ing@gmail.com");
        assignment.setMoved(true);

        var phoneList = new java.util.ArrayList(0);
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40750665523", "type", "1")));
        phoneList.add(AddressBook2.newPhone(DTOFactory.newDTO("number",
                "+40250405090", "type", "2")));

        assignment.setPhones(new java.util.HashSet(phoneList));

        AddressBook2.updateAssignment(assignment);
    } catch (e) {

    }

    var email = assignment.getEmail();

    return " Assignment : " + email;
}

function addAssignment() {
    var phoneList = new java.util.ArrayList(0);
    phoneList.add(DTOFactory.newDTO("number", "+40750665523", "type", "1"));
    phoneList.add(DTOFactory.newDTO("number", "+40250405090", "type", "2"));

    var personId = 75895;
    var organisationId = 71256;
    var dto = DTOFactory.newDTO("personId", personId, "organisationId",
            organisationId, "department", "Software", "position", "Engineer",
            "email", "siatel.software@gmail.com", "phones", phoneList);
    AddressBook2.addAssignment(dto);
}

function deleteAssignment() {
    var personId = 75895;
    var organisationId = 71256;

    AddressBook2.deleteAssignment(personId, organisationId);
}

////////////////////////////////////
//
// Tag
//
```

```
////////////////////////////////////

function createTag() {
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var name = sdf.format(new java.util.Date());

    var tag = AddressBook2.newTag();
    tag.setName("SIATEL_" + name);

    AddressBook2.createTag(tag);

    return "Tag id : " + tag.getId();

}

function createTag2() {

    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var name = "SIATEL_" + sdf.format(new java.util.Date());

    var tag = AddressBook2.newTag(DTOFactory.newDTO("name", name));
    AddressBook2.createTag(tag);

    return "Tag id : " + tag.getId();

}

function updateTag() {
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var today = sdf.format(new java.util.Date());
    var name = "SIATEL_TAG_" + today;

    var id = 73588;

    try {

        var tag = AddressBook2.readTag(id);
        tag.setName(name);

        AddressBook2.updateTag(tag);

        name = tag.getName();

    } catch (e) {

    }

    return "Tag id + name : " + id + " " + name;

}

function addTag1() {
    var contactId = 73512;
    var contactType = "person";
    var contact = AddressBook2.readContact(contactId, contactType);

    var tagId1 = 75673;
    var tag1 = AddressBook2.readTag(tagId1);

    var tagId2 = 67044;
    var tag2 = AddressBook2.readTag(tagId2);

    var tags = new java.util.ArrayList(0);
    tags.add(tag1);
```

```
    tags.add(tag2);

    try {
        AddressBook2.addTag(contact, tags);
    } catch (e) {

    }
}

function addTag2() {
    var contactId = 73512;
    var contactType = "person";
    var contact = AddressBook2.readContact(contactId, contactType);

    var tags = new java.util.ArrayList(0);
    AddressBook2.enumerateTags(null, tags);

    try {
        AddressBook2.addTag(contact, tags);

    } catch (e) {

    }
}

function removePersonTag() {
    var contactId = 73512;
    var tagId = 71316;

    var person = AddressBook2.readContact(contactId, "person");
    var tag = AddressBook2.readTag(tagId);

    try {
        AddressBook2.removeTag(person, tag);

    } catch (e) {

    }
}

function removeOrganisationTag() {
    var contactId = 71317;
    var tagId = 67048;

    var organisation = AddressBook2.readContact(contactId, "organisation");
    var tag = AddressBook2.readTag(tagId);

    try {
        AddressBook2.removeTag(organisation, tag);

    } catch (e) {

    }
}

function deleteTag() {
    var id = 75679;
    try {
        AddressBook2.deleteTag(id);

    } catch (e) {

    }
}
```

```
function idOfTag() {
    var name = "SIATEL";
    var id = AddressBook2.idOfTag(name);

    return "Tag id : " + id;
}

function enumerateTags() {
    var tags = new java.util.ArrayList(0);

    var json = '{ "tagList" : [ ';

    try {

        AddressBook2.enumerateTags(null, tags);

        if (tags != null && !tags.isEmpty()) {

            for (var i = 0; i < tags.size(); i++) {
                var tag = tags.get(i);

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "tagId" : "' + tag.getId()
                        + '", "tagName" : "' + tag.getName() + '"}';

            }
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}

function enumerateTags2() {
    var tags = new java.util.ArrayList(0);

    var json = '{ "tagList" : [ ';

    try {

        var criteria = DTOFactory.newDTO("query", "siatel");

        AddressBook2.enumerateTags(criteria, tags);

        if (tags != null && !tags.isEmpty()) {

            for (var i = 0; i < tags.size(); i++) {
                var tag = tags.get(i);

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "tagId" : "' + tag.getId()
                        + '", "tagName" : "' + tag.getName() + '"}';

            }
```

```
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}

function enumerateTags3() {
    var tags = new java.util.ArrayList(0);

    var json = '{ "tagList" : [ ';

    try {

        var tagIds = new java.util.ArrayList(0);
        tagIds.add("67044");
        tagIds.add("69328");

        var tagNames = new java.util.ArrayList(0);
        tagNames.add("siatel");
        tagNames.add("ing");

        var criteria = DTOFactory
                .newDTO("tagIds", tagIds, "tagNames", tagNames);

        AddressBook2.enumerateTags(criteria, tags);

        if (tags != null && !tags.isEmpty()) {

            for (var i = 0; i < tags.size(); i++) {
                var tag = tags.get(i);

                if (i > 0) {
                    json = json + ', ';
                }

                json = json + '{ "tagId" : "' + tag.getId()
                        + '", "tagName" : "' + tag.getName() + '"}';

            }
        }
    } catch (e) {

    }

    json = json + ' ] }';

    return json;
}
```

SUPPORT POUR UUID POUR LES COMMANDES GEDLIBRARY.READ_FOLDER/DOCUMENT

Si l'objet reçu dans `K.OBJECT` a le numérique principal de l'objet mis à 0, mais le champ `UUID` est saisi, l'objet

(dossier/document/pin/process) sera chargé par `UUID` depuis la base indiquée par le numérique `dbId` depuis l'identifiant d'objet reçu dans `K.OBJECT`.

1. GedLibrary.CMD_REBUILD_ID

- reconstruction d'un identifiant de document à l'aide d'un id incomplet (préfixe, identifiant numérique de la base et identifiant numérique de l'objet)

```
String docId = gedProxy.invoke(conn, GedLibrary.CMD_REBUILD_ID, DTOFactory.newDTO(
                              K.OBJECT_ID, IdFormat.toString(Prefix.FILE_DOCF, nDbId, 0,
0, nDocId)
             )).getString(K.OBJECT_ID);
```

- reconstruction d'un identifiant d'on objet 'viewable' en utilisant le champ `UUID` et un identifiant incomplet (préfixe et identifiant numérique de la base)

```
String docId = gedProxy.invoke(conn, GedLibrary.CMD_REBUILD_ID, DTOFactory.newDTO(
                K.OBJECT_ID, IdFormat.toString(Prefix.FILE_DOCF, nDbId),
                K.OBJECT_UUID_STRING, documentUUIDAsString
             )).getString(K.OBJECT_ID);

String pinId = gedProxy.invoke(conn, GedLibrary.CMD_REBUILD_ID, DTOFactory.newDTO(
                              K.OBJECT_ID, IdFormat.toString(Prefix.PINF, nDbId),
                              K.OBJECT_UUID_STRING, pinUUIDAsString
             )).getString(K.OBJECT_ID);

String folderId = gedProxy.invoke(conn, GedLibrary.CMD_REBUILD_ID, DTOFactory.newDTO(
                              K.OBJECT_ID, IdFormat.toString(Prefix.FOLDER, nDbId),
                              K.OBJECT_UUID_STRING, folderUUIDAsString
             )).getString(K.OBJECT_ID);
```

2. GedLibrary.CMD_READ_ (viewable)

- lecture d'un objet en utilisant le champ `K.OBJECT` (objet complet).
  - fonction standard ou l'identifiant complet est fourni pour la lecture de l'objet

```
Viewable obj = gedProxy.invoke(conn, GedLibrary.CMD_READ_ , DTOFactory.newDTO(
                                    K.OBJECT, SiatelFactory.newI(objectFullId),
                                    K.FLAGS, objReadFlags
                              )).getString(K.OBJECT);
```

- lecture d'on objet à partir du champ `UUID` et un identifiant incomplet (préfixe et identifiant numérique de la base)

```
Viewable obj = gedProxy.invoke(conn, GedLibrary.CMD_READ_ , DTOFactory.newDTO(
                                    K.OBJECT,
SiatelFactory.newI(IdFormat.toString(objPrefix, nDbId),"UUIDString",obj_uuid.toString())
                                    K.FLAGS, objReadFlags
                              )).getString(K.OBJECT);
```

- lecture d'un objet en utilisant le champ `K.OBJECT_ID` (identifiant de l'objet)
  - fonction standard ou l'identifiant complet est fourni pour la lecture de l'objet

```
Viewable obj = gedProxy.invoke(conn, GedLibrary.CMD_READ_ , DTOFactory.newDTO(
                                     K.OBJECT_ID, objectFullId,
                                     K.FLAGS, objReadFlags
                               )).getString(K.OBJECT);
```

- lecture d'on objet à partir du champ `UUID` et un identifiant incomplet (préfixe et identifiant numérique de la base)

```
Viewable obj = gedProxy.invoke(conn, GedLibrary.CMD_READ_ , DTOFactory.newDTO(
                         K.OBJECT_ID, IdFormat.toString(objPrefix, nDbId),
                         K.OBJECT_UUID_STRING, obj_uuid.toString()),
                         K.FLAGS, objReadFlags
                 )).getString(K.OBJECT);
```

Modules complémentaires et autres outils

N° de version des modules complémentaires

Module complémentaire spécifique à Google Chrome

8.9600.0

Module complémentaire commun à tous les navigateurs

8.3.10391

Module complémentaire pour Microsoft Office

8.3.10391

Module complémentaire pour Microsoft Outlook

8.3.10391

Module complémentaire pour Mozilla Thunderbird

8.3.10391

Module complémentaire pour OpenOffice

8.3.10391

Module complémentaire pour LibreOffice

8.3.10391

Module complémentaire pour Zimbra zimlet

8.3.10391

N° des versions des autres outils

Versions des outils disponibles

Intelliscan ver. 8.3.10391

20240422 - R8.3.10391 - IS

Custom Scan [DOSSIERS RH] ver. 8.3.10391

20240422 - R8.3.10391 - Custom Scan [DOSSIERS RH]

EasyScan ver. 8.3.10391

20240422 - R8.3.10391 - EasyScan

Media Tool ver. 8.3.10391

20240422 - R8.3.10391 - Siatel Media Tool

Media Tool Redist ver. 8.3.10391

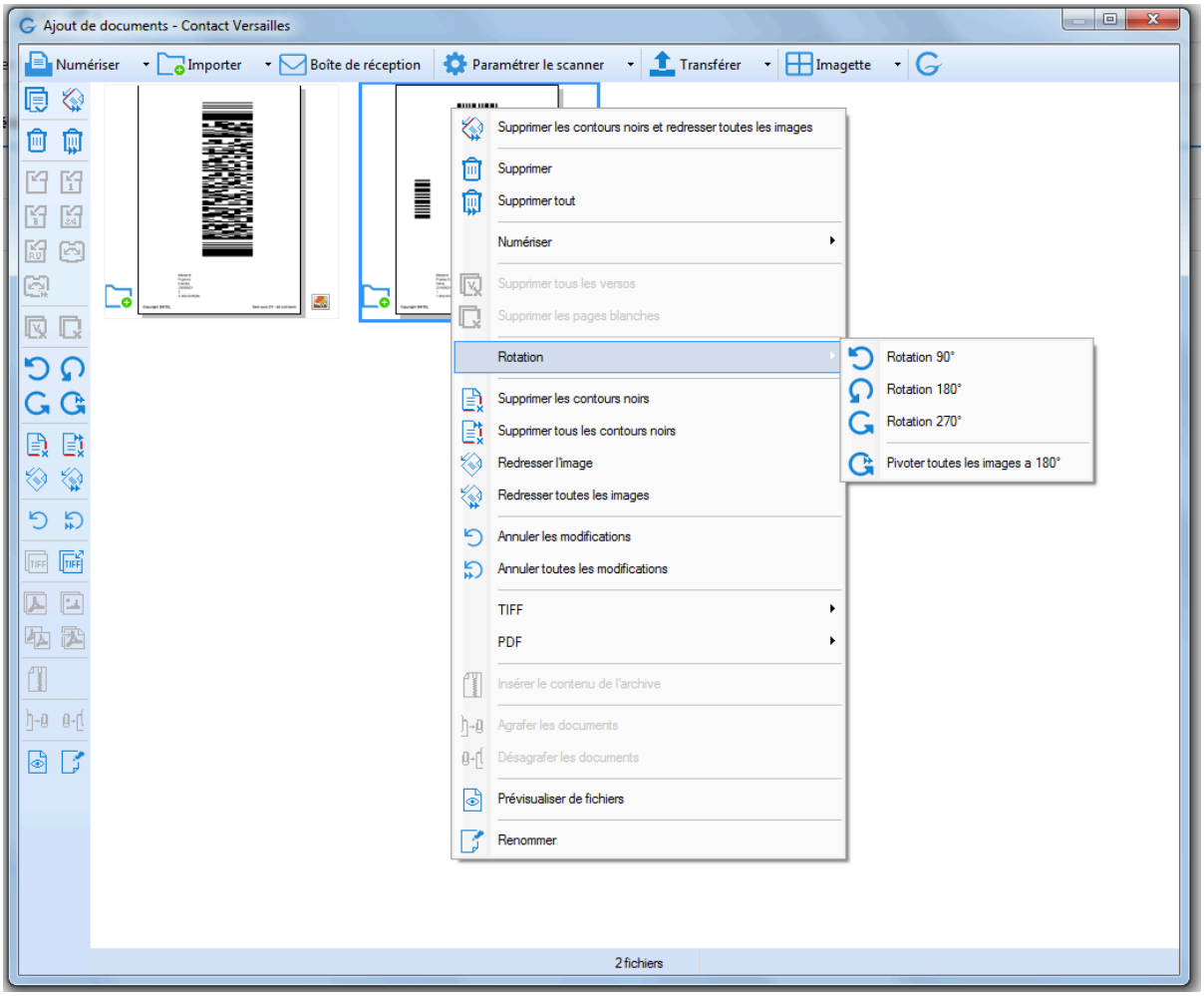
20240422 - R8.3.10391 - Siatel Media Tool Redist

Module complémentaire commun à tous les navigateurs

Interface d'ajout de documents : taille des icônes fonction de la résolution d'écran

Pour les résolutions d'écran inférieures à 1900 px, les icônes qui se trouvent dans la barre d'outils verticale et dans les menus ont une taille de 20x20 px. Pour les autres résolutions, elles ont une taille de 24x24 px.

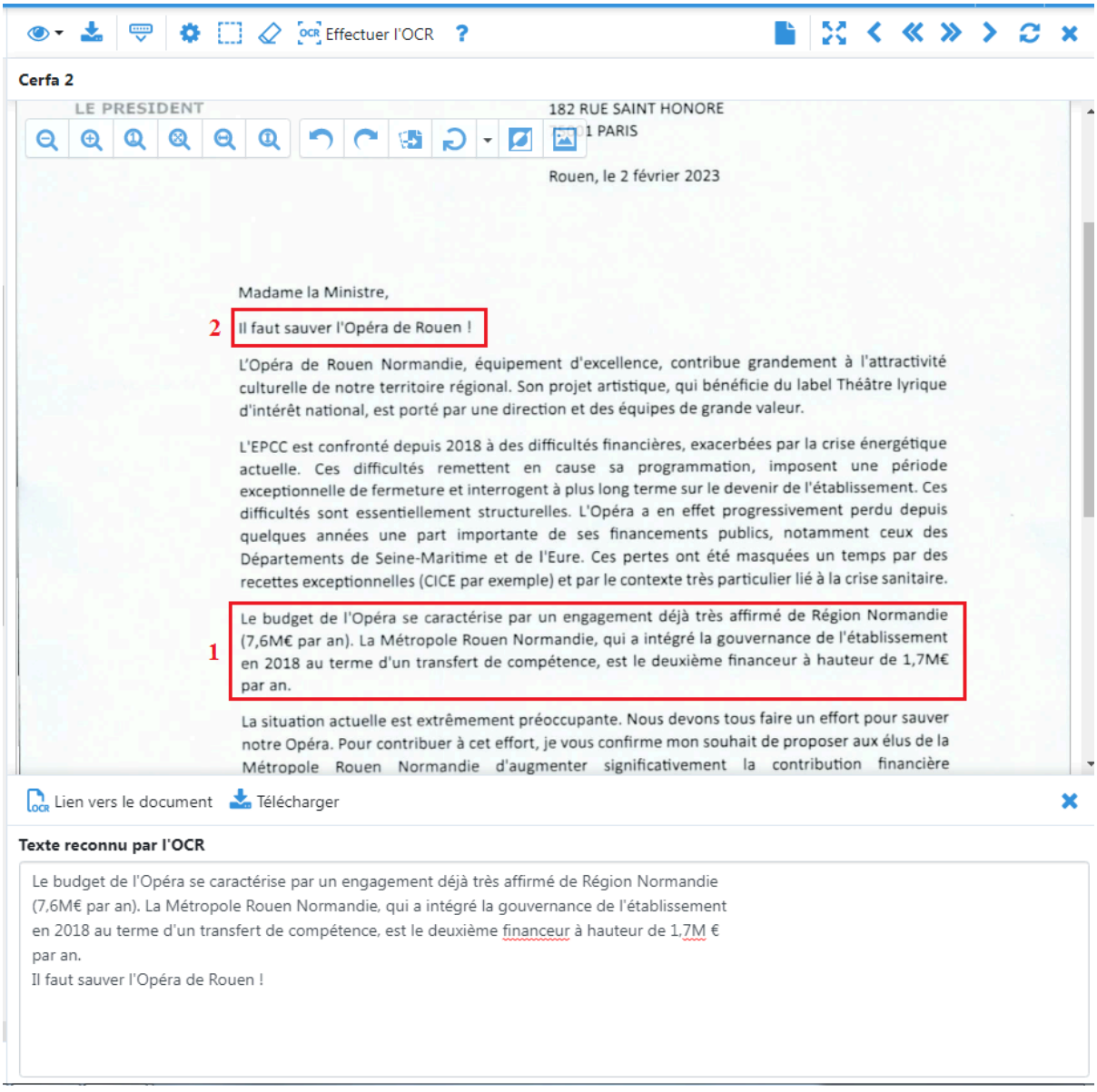
Voici comment se présentent les icônes de 24x24px :



OCR

Prise en compte de l'ordre d'encadrement dans le résultat obtenu

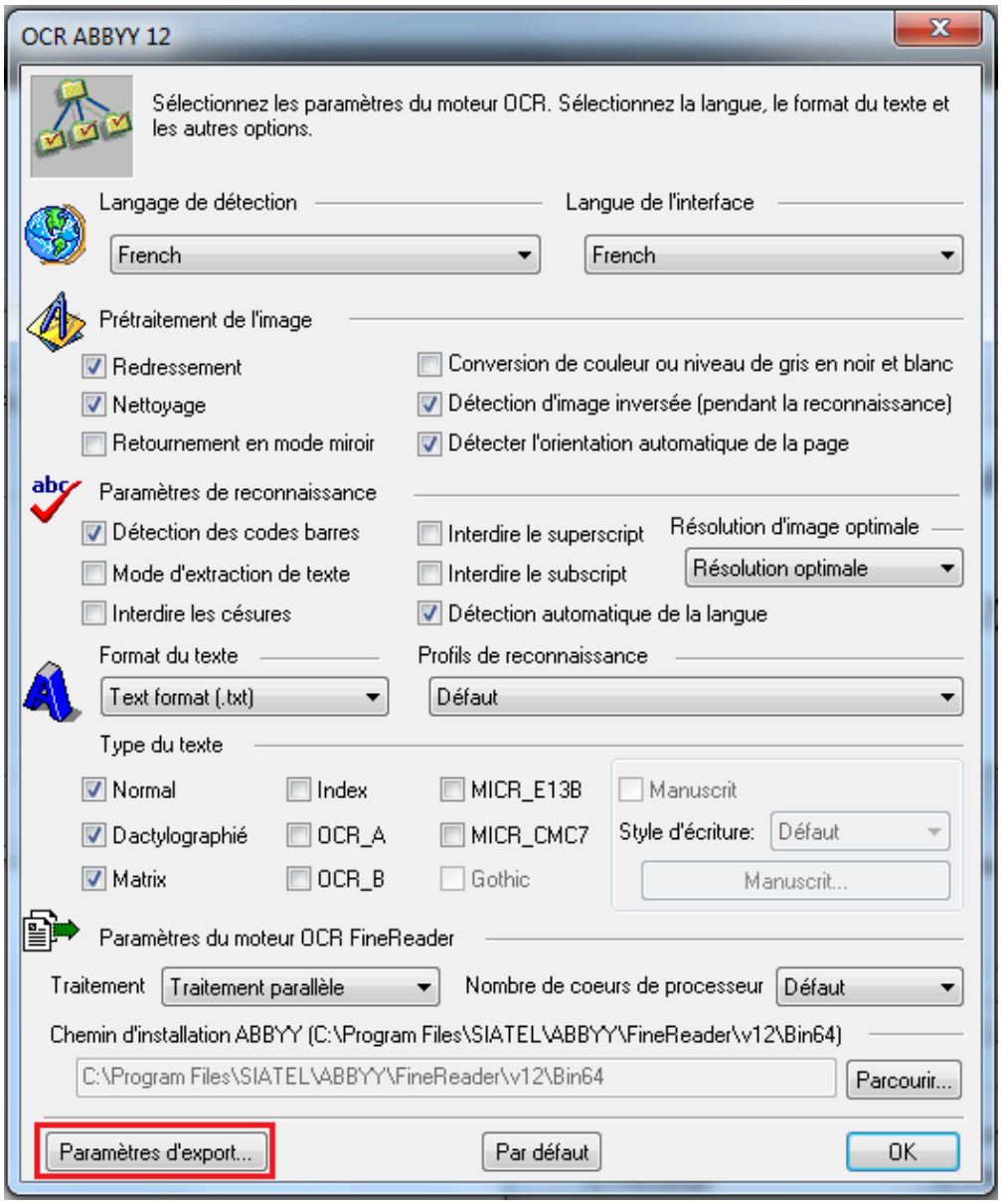
L'ordre dans lequel les zones sont encadrées à l'écran par l'utilisateur détermine la séquence de reconnaissance du texte, page par page. Cette fonctionnalité est prise en charge par les moteurs OCR **Tesseract** et **ABBYY 12**.

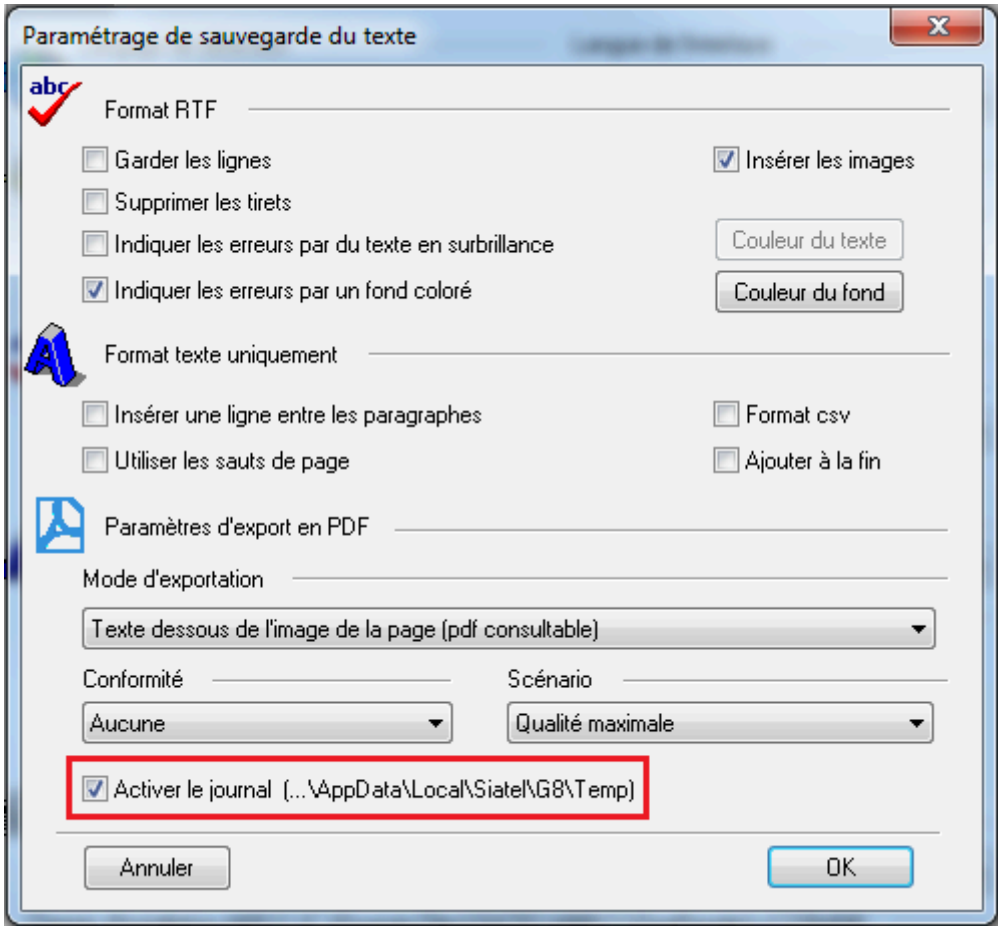


Activation du journal d'événements

L'option **Activer le journal**, accessible depuis le bouton **Paramètres d'export**, permet de déclencher la constitution d'un journal d'événements sur son poste de travail.

Cette option s’applique uniquement au moteur d’OCR ABBYY (10 et 12).

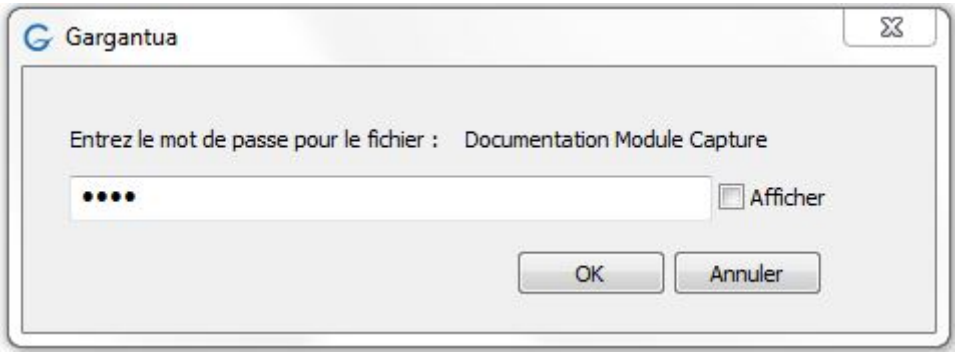

OCR ABBYY 12
Sélectionnez les paramètres du moteur OCR. Sélectionnez la langue, le format du texte et les autres options.
Langage de détection
French
Langue de l'interface
French
Prétraitement de l'image
Redressement
Nettoyage
Retournement en mode miroir
Conversion de couleur ou niveau de gris en noir et blanc
Détection d'image inversée (pendant la reconnaissance)
Détecter l'orientation automatique de la page
Paramètres de reconnaissance
Détection des codes barres
Mode d'extraction de texte
Interdire les césures
Interdire le superscript
Interdire le subscript
Détection automatique de la langue
Résolution d'image optimale
Résolution optimale
Format du texte
Text format (.txt)
Profils de reconnaissance
Défaut
Type du texte
Normal
Dactylographié
Matrix
Index
OCR_A
OCR_B
MICR_E13B
MICR_CMC7
Gothic
Manuscrit
Style d'écriture:
Défaut
Manuscrit...
Paramètres du moteur OCR FineReader
Traitement
Traitement parallèle
Nombre de coeurs de processeur
Défaut
Chemin d'installation ABBYY (C:\Program Files\SIATEL\ABBYY\FineReader\v12\Bin64)
C:\Program Files\SIATEL\ABBYY\FineReader\v12\Bin64
Parcourir...
Paramètres d'export...
Par défaut
OK



Fichiers compressés

Prise en compte des fichiers compressés protégés par un mot de passe

Depuis la version 8.3.10391, il est possible d'extraire le contenu d'un fichier compressé lorsque celui-ci est protégé par un mot de passe. Pour cela, une boîte de dialogue s'ouvre pour en permettre la saisie.



Extraction de fichiers compressés lors de l'importation : formats supportés

Lors de l'importation, Gargantua permet aux utilisateurs d'extraire automatiquement le contenu des fichiers compressés dans tous les formats courants, y compris **.RAR**, **.7z**, et **.ARJ**, en plus du format **.ZIP**.

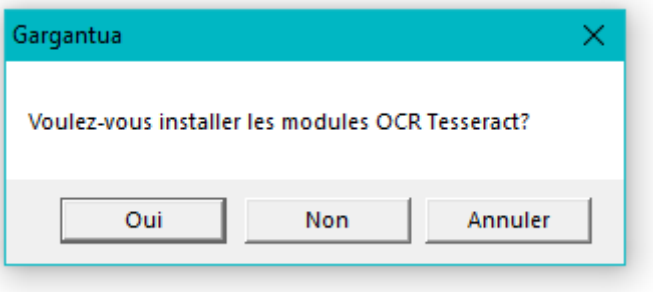
Remarques :

- Pour cette opération d'extraction, il est nécessaire que les modules complémentaires soient installés sur le poste de travail de l'utilisateur.
- Pour rappel, avant la version 8.3.10391, seul le format .ZIP était pris en charge pour cette extraction automatique.

Inclusion de l'installation de l'OCR Tesseract dans l'installation du module complémentaire commun aux navigateurs

Pour faciliter l'installation de l'OCR Tesseract sur les postes de travail, celle-ci a été incluse dans la procédure d'installation du module complémentaire commun à tous les navigateurs.

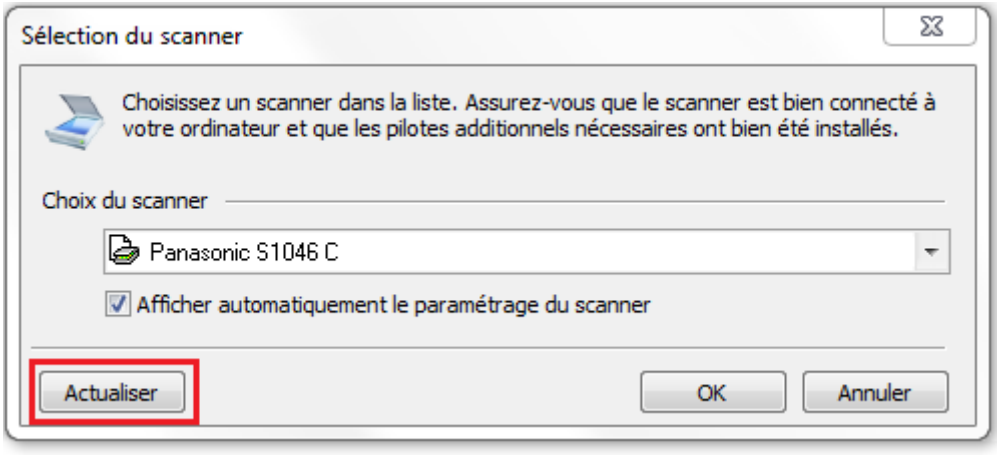
Cette installation, qui est optionnelle, doit être confirmée par l'utilisateur :



Actualisation de la liste des scanners disponibles

Lorsqu'un driver de scanner est installé sur un poste de travail, indépendamment de la procédure d'installation du module complémentaire commun aux navigateurs, alors il est possible que ce scanner n'apparaisse pas dans la liste des scanners pouvant être sélectionnés au moment de la numérisation.

Dans ce cas, il convient d'actualiser cette liste à l'aide du bouton **Actualiser**.

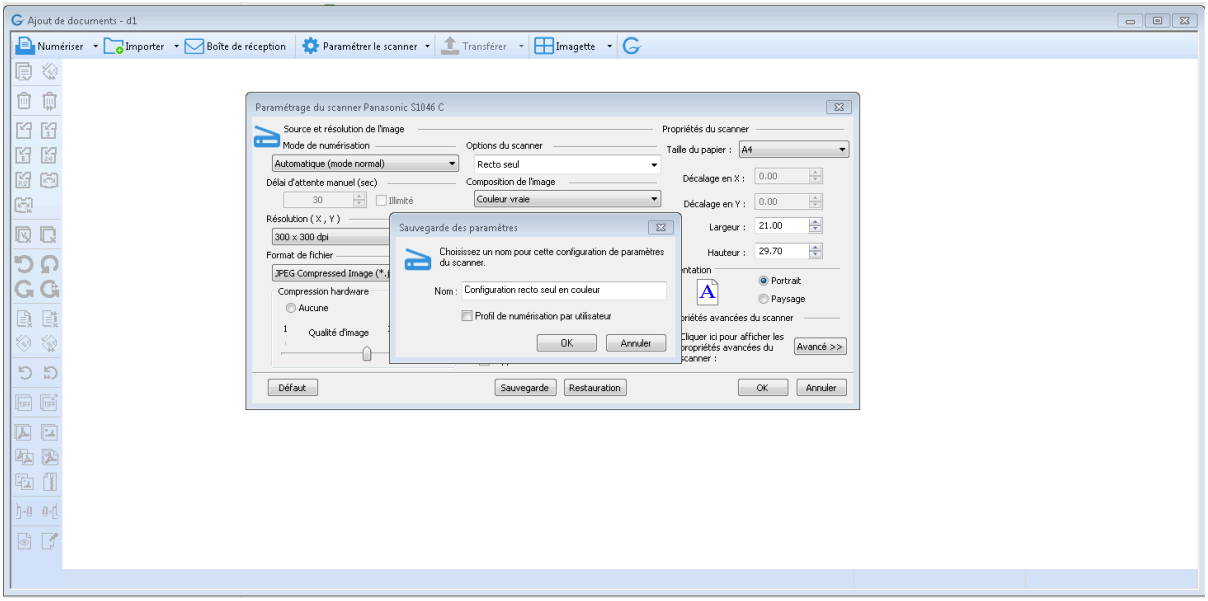


OPTIMISATIONS

Des optimisations ont été apportées pour un chargement plus rapide :

- Des pilotes de scanners
- Des profils de numérisation

> ***Un profil de numérisation correspond à une configuration dans laquelle tous les paramètres d'un scanner ont été sauvegardés.***



Fichiers PDF

Extraction des illustrations contenues dans un fichier PDF

Depuis la version 8.3.10391 du module complémentaire commun à tous les navigateurs, il est possible d'extraire plusieurs illustrations par page, par défaut, jusqu'à 5.

Par défaut également, les images de dimensions inférieures à 15px sont ignorées.

Pour changer ces valeurs par défaut, il faut modifier les entrées de registre sur le poste de travail :

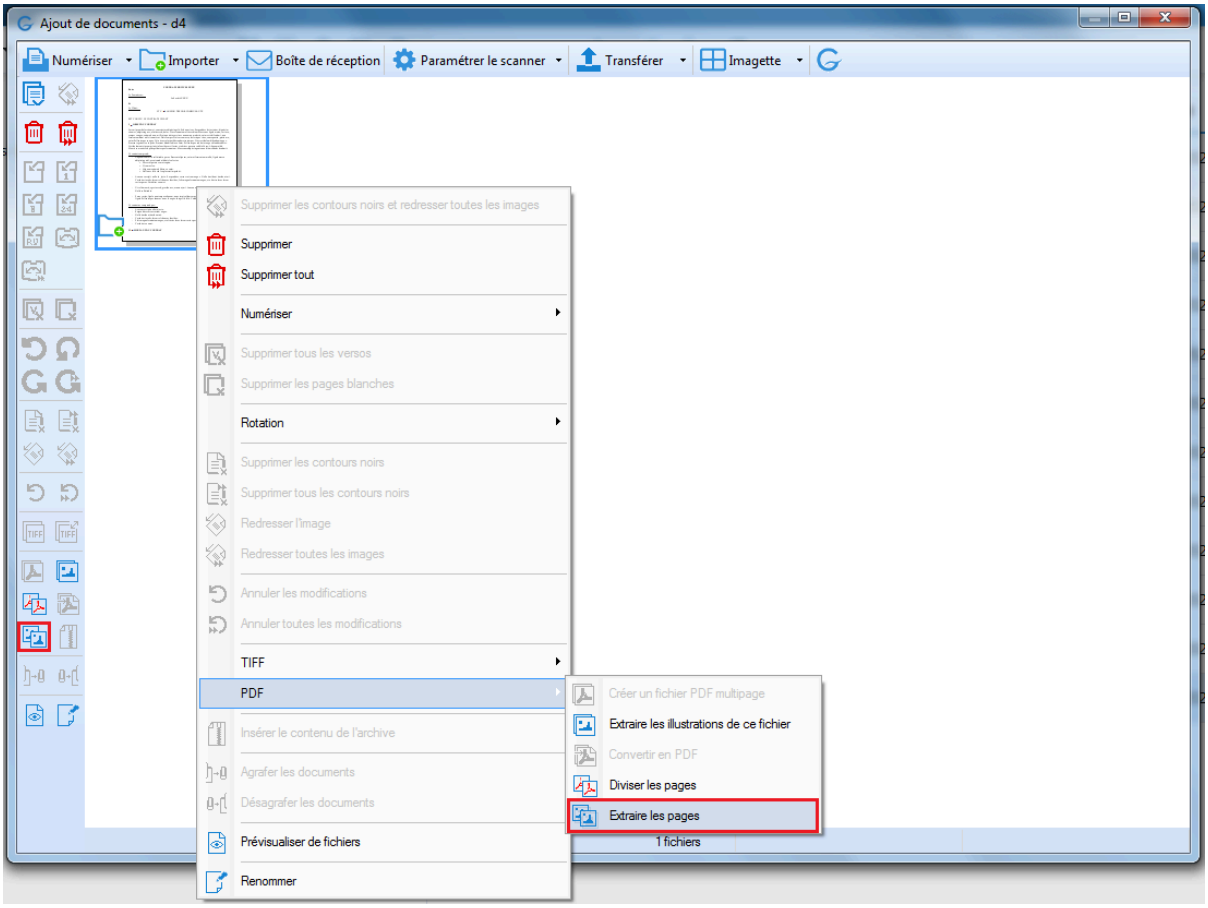
HKEY_CURRENT_USER\Software\SIATEL\Gargantua 8\Settings\Podofo :

- **MaxNoImagesPerPage** : 5
- **MinImageDimension** : 15

Commande d'extraction des pages en images

La version 8.3.10391 amène la commande **Extraire les pages** pour le traitement des fichiers PDF.

Cette commande extrait chaque page d'un fichier PDF et la sauvegarde en tant qu'image au format PNG. On obtient ainsi autant fichiers images que le fichier PDF contient de pages.



Pour rappel, les autres commandes sur les fichiers pdf sont :

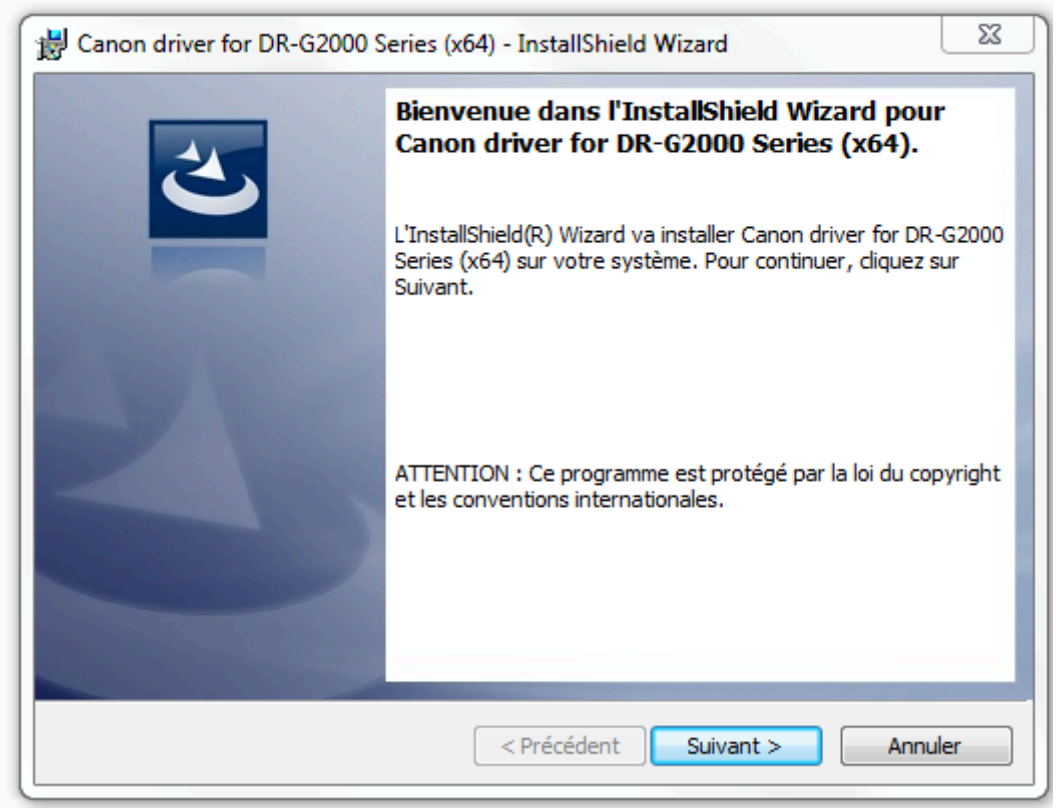
- **Extraire les illustrations** d'un fichier PDF : extrait les illustrations existantes dans le fichier pdf. Une page du fichier pdf peut contenir plusieurs illustrations. Voir le chapitre précédent.
- **Diviser les pages** : qui divise le fichier pdf en autant fichiers pdf que de pages dans le fichier PDF initial. Les fichiers pdf obtenus ont une seule page.

Dans le **Module Acquisition de documents**, cette opération a remplacé l'ancienne modalité de division de pages pour les fichiers PDF dans les lots avec l'option PDF « Découpe par page » avec « Eclater le document ». Les fichiers png obtenus doivent avoir la résolution définie dans le lot.

Compatibilité avec le driver du scanner Canon DR-G2090

A partir de la version 8.3.10391 du module complémentaire commun à tous les navigateurs, le scanner Canon DR-G2090 peut être utilisé pour la numérisation de documents.

Prérequis : Le driver fourni par le fabricant pour les scanners DR-G2000 doit être installé sur l'ordinateur.



Limitations et problèmes connus :

- La détection de feuilles de séparation ne fonctionne que dans le mode de numérisation sans batch, page par page (mode sans numérisation rapide).
- Lors de la détection de feuilles en double, la numérisation s'arrête définitivement quand il y a peu de feuilles dans l'introducteur à cause du fait que l'introducteur descend. Le même problème est observé pour la numérisation de paquets avec des paramétrages différents.
- En mode de numérisation 'multistream', qui combine des images en couleur avec des images en noir et blanc ou en niveaux de gris, toutes les pages sont d'abord numérisées en couleur. Les versions en noir et blanc ou en niveaux de gris sont ensuite créées par un traitement logiciel. Ce processus supplémentaire de conversion par logiciel ralentit la vitesse de numérisation.
- La détection automatique de couleur lors de la numérisation ralentit le processus, car chaque page est initialement numérisée en couleur. Si le document est ensuite identifié comme étant en noir et blanc ou en niveaux de gris, une conversion est effectuée par le logiciel, ce qui diminue la vitesse globale de numérisation.

Ces limitations s'appliquent pour tous les outils incluant le driver du scanner Canon DR-G2090.

IntelliScan

Compatibilité avec le driver du scanner Canon DR-G2090

A partir de la version 8.3.10391 de l'outil IntelliScan, le scanner Canon DR-G2090 peut être utilisé pour la numérisation dans les tâches IntelliScan.

Prérequis : le driver fourni par le fabricant pour les scanners DR-G2000 doit être installé sur l'ordinateur.

EasyScan

Compatibilité avec le driver du scanner Canon DR-G2090

A partir de la version 8.3.10391 de EasyScan, le scanner Canon DR-G2090 peut être utilisé pour la numérisation dans les scripts EasyScan.

Prérequis : le driver fourni par le fabricant pour les scanners DR-G2000 doit être installé sur l'ordinateur.

CustomScan

Compatibilité avec le driver du scanner Canon DR-G2090

A partir de la version 8.3.10391 de l'outil, le scanner Canon DR-G2090 peut être utilisé pour la numérisation dans les scripts CustomScan.

Prérequis : le driver fourni par le fabricant pour les scanners DR-G2000 doit être installé sur l'ordinateur.

Modules complémentaires bureautiques

Fonction d'affichage de document disponible depuis un formulaire

La fonction **form_view_document** est disponible dans les formulaires utilisés dans les différents modules complémentaires bureautiques.

Aucune autre commande appelée depuis le formulaire ne fonctionnera dans ce cas (ajouter, numériser, créer, diviser un PDF, signer, modifier, modifier nativement, etc.). De plus, certaines options de l'aperçu ont été supprimées implicitement pour éviter des problèmes : le bouton d'ouverture dans l'application d'origine, le bouton de passage en mode OCR, le panneau de liens et les commandes de transformation d'image n'apparaîtront pas.

Extensions/Modules serveur

Module Moteur de classement

Configuration d'une temporisation

Un paramètre permet de définir une valeur en secondes qui sert de temporisation pour décaler le démarrage des processus successifs. Il a été mis en place pour éviter la génération de doublons.

autoimport.classification.[N].process.first.seconds = [nombre entier]

[nombre entier] : décalage souhaité exprimé en secondes

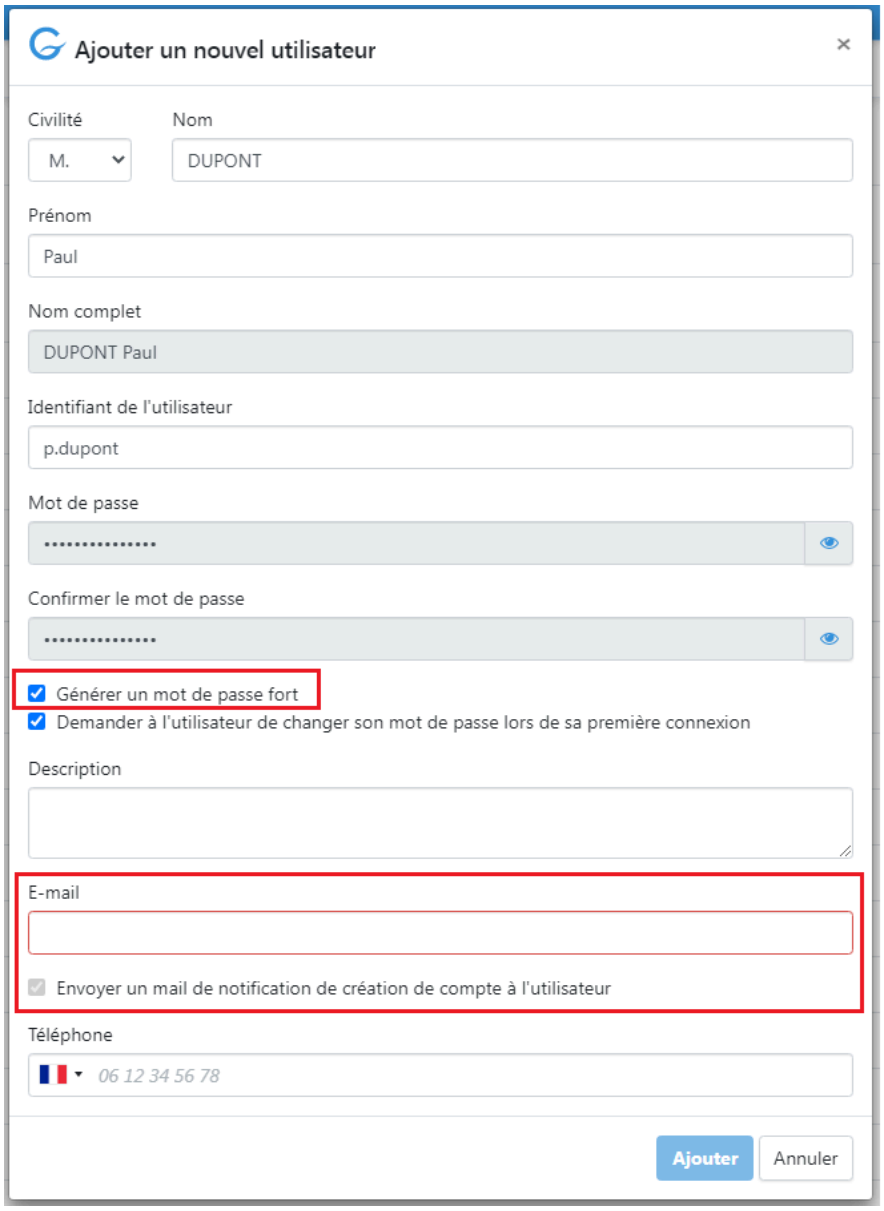
Module Gestionnaire

Changement du propriétaire

Une nouvelle commande a été apportée dans la gestion des utilisateurs : **Changer le propriétaire**.

Génération d'un mot de passe fort aléatoire

Depuis la version 8.3, la fenêtre de création d'un utilisateur a été enrichie d'une nouvelle propriété permettant de générer un mot de passe fort de manière aléatoire. Si cette option est choisie, l'adresse email et l'envoi d'une notification de création du compte deviennent obligatoires.



Autres développements

Serveur OCR

Une nouvelle version de serveur OCR est disponible : **20240418 - R8.3.15 - SERVER OCR**.

Détection automatique de la langue d'un document

La détection de la langue du texte reconnu par l'OCR permet d'identifier la/les langue(s) d'un document.

La langue principale du document est détectée dans le texte reconnu, mais également la collection d'autres langues trouvées dans le document.

Les langues de la collection sont triées par fréquence d'occurrence : de la plus fréquente à la moins fréquente.

La réponse du serveur est transmise au format JSON :

```
{
  « op » : « ocrserver »,
  « status » : 0/1/2/3,
  « message » : « success », / « error string »,
  « result » :
  {
    « 1 »: « texte reconnu dans la page 1 »,
    « 2 » : « texte reconnu dans la page 2 »,
     ...
    « n » : « texte reconnu dans la page n »
  },
  « langs » :
  {
  « basic »: « French »,
  « detected » : [ »English, German »] // ISO language name
  }
}
```

- « result » : s'indique pour l'OCR par zones.
- « basic » = La langue principale identifiée sur le document
- « detected » = La collection de langues détectées dans le document analysé
- « status » = 0 - success, 1 - error

Cette fonctionnalité n'est disponible que pour les serveurs qui utilisent ABBYY 12.

Cette fonctionnalité est disponible à partir de la version 8.2.13 du serveur OCR.

Méthodes de détection de langue

Deux méthodes de détection de la langue d'un document sont disponibles :

- Détection uniquement de la langue du document, sans la reconnaissance du texte.
- Extraction du texte reconnu par OCR avec détection de la langue du document.

Ces deux méthodes correspondent aux options de l'extension **Connexion au serveur OCR.** (voir aussi le chapitre dédié).

Détection uniquement de la langue du document

Dans l'arborescence des dossiers du traitement de documents du serveur OCR (**..\SIATEL\SERVICES**), à la première utilisation de la détection de langue, un nouveau dossier **DETECT** est créé. Il sera utilisé pour faire transiter les documents provenant du serveur Gargantua pour lesquels seule la langue du document est détectée. L'extension **Connexion au serveur OCR** utilise ce dossier pour envoyer au serveur OCR les documents pour lesquels uniquement la détection de la langue est demandée.
La détection de langue s'effectue également si la commande JSON envoyée depuis Gargantua contient la propriété { »params »: « detectLang » : « 1 »).
Dans ce cas, la détection de langue se fait en chargeant et recherchant toutes les langues prises en charge par ABBYY12. Cette opération peut prendre beaucoup de temps.
Pour cette raison, dans le fichier de configuration du serveur OCR, une propriété permet désormais de spécifier pour que la détection se fasse soit sur la première page, soit sur le document entier :

<**detect.languages.mode**> = 1/0 (ou 0 - document entier, 1 - première page)

Extraction du texte reconnu par OCR avec détection de la langue du document

Si le serveur OCR doit extraire le texte et simultanément détecter la langue, seules les langues définies dans le fichier de configuration du serveur OCR seront chargées. La langue détectée sera déterminée à partir de cette liste.

Pour rappel, voir l'option de configuration du serveur OCR : **<languages>** = French;English;German;Spanish;Digits;Romanian;...
Le résultat de la détection de langue est un JSON au format suivant :

```
'langs' :
{
  « basic »: « fra »,
```

« confidence » : « HIGH »,
« detected » : [»eng », « deu »]
}

- ‘basic’ : La langue principale identifiée dans le texte du document
- ‘confidence’ : HIGH / MEDIUM / LOW / NONE
- ‘detected’ : La collection de langues détectées dans le document respectif.

Langues reconnues et les codes utilisés

Les langues détectées sont en format **ISO 639-3**, qui est une nomenclature normalisée utilisée pour classer les langues. Voir
https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes pour plus d'informations.
Les langues reconnues par ABBYY 12, à partir desquelles la langue du texte sera détectée, ainsi que les codes ISO963-3 associés, sont :

Pays	ISO 639-3
Abkhaz	abk
Adyghe	ady
Afrikaans	afr
Agul	agx
Albanian	sqi
Altaic	tut
Arabic (Saudi Arabia)	ara
Armenian Eastern	hye
Armenian Western	hyw
Armenian Grabar	xcl
Avar	ava
Aymara	aym
Azerbaijani (Cyrillic)	aze
Azerbaijani (Latin)	aze
Bashkir	bak
Basque	eus

Belarussian	bel
Bemba	bem
Blackfoot	bla
Breton	bre
Bugotu	bgt
Bulgarian	bul
Buryat	bua
Catalan	cat
Chamorro	cha
Chechen	che
Chinese Simplified	zho
Chinese Traditional	zho
Chukcha	ckt
Chuvash	chv
Corsican	cos
Crimean Tatar	crh
Croatian	hrv
Crow	cro
Czech	ces
Danish	dan
Dargwa	dar
Dungan	dng
Dutch (Netherlands)	nld
Dutch (Belgian)	vls
English	eng
Eskimo (Cyrillic)	esx
Eskimo (Latin)	esx
Esperanto	epo

Estonian	est
Even	eve
Evenki	evn
Faeroese	fao
Fijian	fij
Finnish	fin
French	fra
Frisian	fry
Friulian	fur
Scottish Gaelic	gla
Gagauz	gag
Galician	glg
Ganda	lug
German	deu
German (New Spelling)	deu
German (Luxembourg)	ltz
Greek	ell
Guarani	grn
Hani	hni
Hausa	hau
Hawaiian	haw
Hebrew	heb
Hungarian	hun
Icelandic	isl
Ido	ido
Indonesian	ind
Ingush	inh
Interlingua	ina

Irish	gle
Italian	ita
Japanese	jpn
Japanese Modern	jpn
Kabardian	kbd
Kalmyk	xal
Karachay-Balkar	krc
Karakalpak	kaa
Kasub	csb
Kawa	vwa
Kazakh	kaz
Khakas	kjh
Khanty	kca
Kikuyu	kik
Kirgiz	kir
Kongo	kon
Korean	kor
Korean (Hangul)	kor
Koryak	kpy
Kpelle	kpe
Kumyk	kum
Kurdish	kur
Lak	lbe
Sami (Lappish)	smi
Latin	lat
Latvian	lav
Latvian (Gothic)	lav
Lezgin	lez

Lithuanian	lit
Luba	lub
Macedonian	mkd
Malagasy	mlg
Malay	msa
Malinke	mku
Maltese	mlt
Mansi	mns
Maori	mri
Mari	chm
Maya	myn
Miao	hmn
Minankabaw	min
Mohawk	moh
RomanianMoldavia	ron
Mongol	mon
Mordvin	rus
Nahuatl	nhe
Nenets	yrk
Nivkh	niv
Nogay	nog
Norwegian	nor
Norwegian (Bokmal)	nob
Norwegian (Nynorsk)	nno
Nyanja	nya
Occidental	ile
Ojibway	oji
Old English	ang

Old French	fro
Old German	goh
Old Italian	ita
Old Spanish	osp
Ossetic	oss
Papiamento	pap
Tok Pisin	tpi
Polish	pol
Portuguese (Portugal)	por
Portuguese (Brazilian)	por
Provencal	fra
Quechua	que
Rhaeto-Romanic	ita
Romanian	ron
Romanian (Moldavia)	ron
Romany	rom
Ruanda	kin
Rundi	run
Russian (old spelling)	orv
Russian	rus
Samoan	smo
Selkup	sel
Serbian (Cyrillic)	srb
Serbian (Latin)	srb
Shona	sna
Sioux (Dakota)	dak
Slovak	slk
Slovenian	slv

Somali	som
Sorbian	wen
Sotho	sot
Spanish	spa
Sunda	sun
Swahili	swh
Swazi	ssw
Swedish	swe
Tabassaran	tab
Tagalog	tgl
Tahitian	tah
Tajik	tgk
Tatar	tat
Thai	tha
Jingpo	kac
Tongan	ton
Tswana	tsn
Tun	tun
Turkish	tur
Turkmen	tuk
Tuvin	tyv
Udmurt	udm
Uighur (Cyrillic)	uig
Uighur (Latin)	uig
Ukrainian	ukr
Uzbek (Cyrillic)	uzb
Uzbek (Latin)	uzb
Visayan	fil

Vietnamese	vie
Welsh	cym
Wolof	wol
Xhosa	xho
Yakut	sah
Yiddish	yid
Zapotec	zap
Zulu	zul

Détails pour les développeurs

Dans le dossier **DOWNLOAD** (du serveur OCR), un fichier JSON du même nom que le document est créé avec la structure suivante :

```
{
  « op »: »ocrserver »,
  « status »: 0/1/2/3,
  « message »: « success », / « error string »,
  « result » :
  {
    « 1 »: « texte reconnu dans la page 1 »,
    « 2 » : « texte reconnu dans la page 2 »,
     ...
    « n » : « texte reconnu dans la page n »
  },
  « langs » :
  {
  « basic »: « French »,
   »detected » : [ »English, German »] // ISO language name
  }
}
```

Nettoyage des fichiers orphelins après un temps prédéterminé

Le serveur OCR peut laisser des fichiers orphelins sur le disque. Pour y remédier, une nouvelle option de configuration **fichiers.clean.timeout** a été introduite, permettant le nettoyage automatique de ces fichiers après un délai spécifié.

fichiers.clean.timeout = 30 (jours)

Par défaut, ce délai est de 30 jours, calculé à partir de la date de création des fichiers. Une valeur de 0 désactive cette option.

Les fichiers orphelins sont désormais supprimés automatiquement après le délai défini, avec une vérification quotidienne pour éliminer tout fichier créé il y a plus de X jours.

Ce nettoyage se fait aussi au démarrage du service du serveur OCR.

La fonctionnalité est disponible à partir de la version 8.3.15 du serveur OCR.

Prise en compte de l'ordre des zones

Lors de la reconnaissance des zones OCR, l'ordre dans lequel elles ont été sélectionnées correspond au résultat OCR.

L'ordre des zones est respecté sur une même page.

La fonctionnalité est disponible pour les serveurs OCR qui utilisent le moteur OCR ABBYY 12.

Setup d'installation de l'OCR Tesseract

Mise à jour du setup de l'OCR Tesseract à la version 5.3

Un nouveau setup d'installation OCR Tesseract est disponible **20240416 - R8.3.0 - TESSERACT OCR LIGHT**.

Ce setup inclut la version OCR Tesseract 5.3 & Leptonica 1.83.1.

Installation optionnelle d'autres langues que celles par défaut

Pour utiliser d'autres langues que celles prises en charge par défaut par Tesseract, il faut créer un dossier **tesseract** dans le chemin d'installation du serveur Gargantua <SIATEL_HOME>, puis y copier les fichiers de langues souhaitées.

Tous les fichiers de langues disponibles se trouvent dans le package d'installation Gargantua, dans le dossier **Distribution\Tesseract** .

Au niveau du paramétrage du moteur OCR Tesseract, celui-ci essaie de télécharger les fichiers de langue :

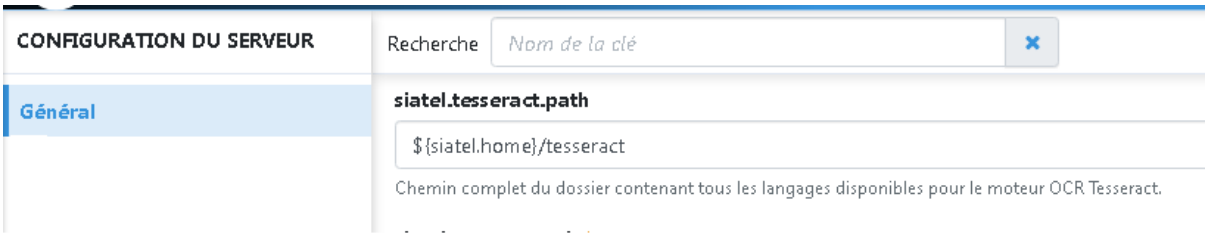
- Qui se trouvent sur le serveur Gargantua
- Qui sont à disposition sur le site web de Tesseract lorsqu'ils ne sont pas sur le serveur Gargantua

(https://raw.githubusercontent.com/tesseract-ocr/tessdata/main/fra.traineddata).

- Si, lors de ces deux tentatives, les fichiers de la langue sélectionnée ne peuvent être téléchargés, alors un message d'erreur s'affiche.

Si aucune langue n'est installée, le moteur OCR Tesseract essaie de télécharger l'anglais, le français, l'allemand, l'espagnol et l'italien, qui sont les langues gérées **par défaut**.

Le chemin complet du dossier contenant les langues disponibles pour le moteur OCR Tesseract est paramétrable dans la section **Général**, dans la **Configuration du serveur**.



Mise à jour de la librairie PDFBox

Depuis la version 8.3, la librairie PDFBox a été mise à jour de la version 2.0.25 à la version 20.30.

Cette librairie est utilisée pour certains fonctionnalités comme : la signature de documents, la génération de fichiers PDF, l'extraction des pages/images d'un fichier PDF, la fusion et la division de fichiers PDF.

La librairie est utilisée dans les modules/extensions suivants : **Acquisition de documents** (siatel-plugin-capture), **Module d'extensions personnalisées** (siatel-custom-extensions), **Convertisseurs de documents ImageMagick** (siatel-plugin-cvtimagik), **OCR Serveur** (siatel-plugin-ocrserver), **Sauvegarde des messages depuis Microsoft Outlook/Thunderbird** (siatel-plugin-sendto), **Data extractor plugin** (siatel-plugin-dataextractor), **Listing des documents CD** (siatel-plugin-mindef), **OCR bridge plugin**(siatel-plugin-ocrbridge).