

Version 8.5.11282

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 5

Interface utilisateur 8
Module Portails 8
Widget Liste des processus 8
Module Tâches 9
Améliorations dans l'interface 9
Affichage de la tâche utilisateur suivante après le démarrage automatique d'un processus 9
Module Circuits de signature 11
Paramétrage du circuit 11
Signataires externes 12

Console d'administration 15
Gestion des utilisateurs 15
Définir un utilisateur titulaire pour un rôle fonctionnel 15
Gestion des rôles 16
Utilisateur titulaire pour un rôle fonctionnel 16
Gestion des bases documentaires 16
Gestion des types 16
Gestion des règles de composition 18
Gestion des processus 19
Gestion des scripts 21
Gestion des portails 23
Affichage de la priorité dans la liste des portails 23
Convertisseurs 24
Personnalisation du convertisseur d'usage général par défaut 24
Généralités 26
Indication des extensions manquantes en cas d'indisponibilité du module 26
Outil de gestion les traductions 27
Informations complémentaires dans le bandeau notifiant la lecture seule 31

Modules complémentaires et autres outils 33
N° de version des modules complémentaires 33
N° des versions des autres outils 33
Mise à jour du Microsoft Edge WebView2 vers la dernière version 34

EXTENSIONS/MODULES SERVEUR 37
Module Workflex 37
Bibliothèque de règles 37

AUTRES DÉVELOPPEMENTS 51
API scripting formulaires 51
Support pour télécharger des documents depuis un formulaire 51
Support pour afficher/cacher les colonnes d'une grille 52
APIs scripting serveur 53
Support pour cacher les champs personnalisés dans les circuits de signature 53

INTRODUCTION

Gargantua 8.5.11282 - Une version qui marque un tournant

Nous sommes heureux de vous dévoiler GARGANTUA 8.5.11281, une version qui franchit une nouvelle étape dans l'évolution de la plateforme.

Cette édition a été pensée pour offrir une expérience plus fluide, plus performante et encore mieux alignée avec les besoins opérationnels des organisations modernes.

Avec cette mise à jour, GARGANTUA continue d'affirmer sa position comme solution de référence, en combinant innovation fonctionnelle, robustesse technique et exigence de qualité.

Une version riche en nouveautés

Cette nouvelle édition introduit un ensemble de fonctionnalités inédites, conçues pour simplifier le quotidien des utilisateurs, renforcer la productivité et offrir davantage de flexibilité dans la configuration et l'exploitation de la plateforme.

Vous découvrirez notamment :

- des outils plus intelligents pour optimiser vos processus,
- des interfaces enrichies pour une navigation plus intuitive,
- des mécanismes améliorés pour renforcer la cohérence et la fiabilité de vos données,
- ainsi que de nouvelles possibilités de personnalisation pour adapter GARGANTUA à vos usages spécifiques.

Chaque nouveauté a été pensée pour apporter une valeur immédiate, tout en préparant le terrain pour les évolutions futures.

Des améliorations techniques au service de la performance

Au-delà des nouvelles fonctionnalités, cette version intègre de nombreuses optimisations techniques visant à renforcer la stabilité, la rapidité et la sécurité de la plateforme.

Ces améliorations se traduisent par :

- une meilleure réactivité des interfaces,
- une gestion plus efficace des ressources,
- une compatibilité renforcée avec les environnements modernes,
- et une expérience utilisateur plus homogène, même dans les contextes les plus exigeants.

Corrections et fiabilité renforcée

Comme pour chaque version, vous trouverez également :

- la liste complète des corrections intégrées,

- ainsi que la description des principaux problèmes résolus.

Ces correctifs contribuent à garantir un fonctionnement plus stable et plus prévisible, tout en répondant aux retours et besoins exprimés par les utilisateurs.

Une version pensée pour vous

GARGANTUA 8.5.11281 n'est pas seulement une mise à jour : c'est une nouvelle impulsion.

Elle reflète notre engagement constant à vous offrir une plateforme :

- plus performante,
- plus intuitive,
- plus évolutive,
- et toujours plus adaptée à vos enjeux métiers.

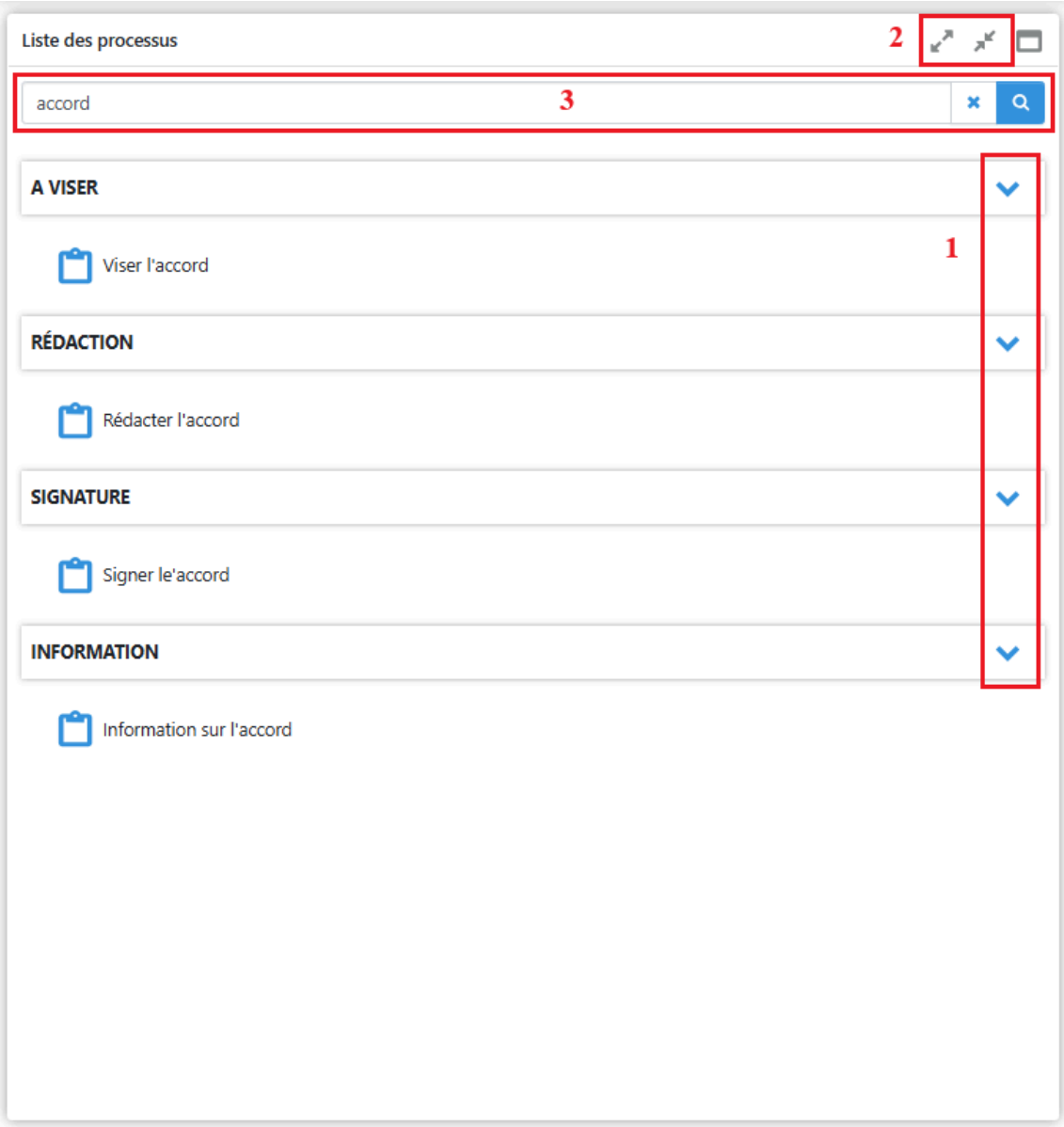
Nous sommes impatients de vous faire découvrir l'ensemble des améliorations apportées dans cette version et de vous accompagner dans leur mise en œuvre.

INTERFACE UTILISATEUR

MODULE PORTAILS

WIDGET LISTE DES PROCESSUS

Depuis la version 8.5.11264, le widget **Liste des processus** a bénéficié d'une série d'améliorations destinées à rendre la navigation plus intuitive et à faciliter l'analyse des processus affichés.



Ces évolutions visent autant le confort d'utilisation que l'efficacité lors de l'exploration de listes complexes. :

- Gestion avancée des groupes de points de démarrage

  Les groupes peuvent désormais être pliés ou dépliés individuellement. Cette nouvelle possibilité permet de réduire l'encombrement visuel lorsque de nombreux processus sont présents, et d'accéder plus rapidement aux sections pertinentes. Cette fonctionnalité est illustrée par l'élément (1) dans la figure.

- Commande globale de pliage/dépliage

  Une nouvelle action a été ajoutée pour plier ou déplier l'ensemble des groupes en un seul clic. Cette commande est particulièrement utile lorsque l'on souhaite réinitialiser rapidement l'affichage ou obtenir une vue d'ensemble structurée. Elle correspond à l'élément (2) dans la figure.

- Ajout d'un champ de recherche intégré

  Pour accélérer la localisation d'un processus spécifique, un champ de recherche a été introduit. Il suffit désormais de saisir une partie du nom du processus pour filtrer instantanément la liste et retrouver l'élément recherché. Cette amélioration est représentée par l'élément (3) dans la figure.

Module Tâches

Améliorations dans l'interface

Depuis la version 8.5.11264, l'interface du module **Tâches** a été enrichie afin d'améliorer la lisibilité et de rendre l'utilisation plus intuitive. Deux évolutions principales ont été apportées :

- Renommage des boutons des tâches en brouillon

  Les libellés des boutons associés aux tâches en brouillon ont été revus pour être plus explicites. Cette mise à jour vise à réduire toute ambiguïté et à permettre aux utilisateurs de comprendre immédiatement l'action associée à chaque bouton, même lors d'une première utilisation.

- Réorganisation de la disposition des boutons

  La disposition des boutons, aussi bien dans les brouillons que dans les tâches actives, a été repensée. L'objectif est d'offrir une expérience plus cohérente et plus naturelle, en plaçant les actions les plus courantes à des emplacements plus accessibles et en harmonisant la présentation globale de l'interface.

Affichage de la tâche utilisateur suivante après le démarrage automatique d'un processus

Depuis la version 8.5.11269, le comportement de l'interface web lors du démarrage d'un processus a été amélioré afin d'offrir une transition plus fluide vers la première tâche utilisateur.

Vérification automatique de l'accès à la première tâche utilisateur

Lorsque l'option d'affichage automatique de la première tâche utilisateur est activée, l'interface vérifie si le processus a atteint une tâche utilisateur accessible par l'utilisateur courant.

Cette vérification est effectuée jusqu'à 15 fois, avec un intervalle de 3 secondes entre chaque tentative.

Comportement lorsque la tâche devient disponible

Dès qu'une tâche utilisateur accessible est identifiée :

- Si le processus a été lancé depuis le portail, l'utilisateur est automatiquement redirigé vers sa boîte de tâches.
- Si le processus a été lancé depuis la boîte de tâches, le filtre approprié est automatiquement sélectionné et la liste des tâches est rafraîchie.

Dans les deux cas, la tâche identifiée est sélectionnée et affichée automatiquement, permettant à l'utilisateur de commencer immédiatement son travail.

Si aucune tâche utilisateur n'est trouvée

Si, après les 15 tentatives, aucune tâche utilisateur accessible n'est détectée :

- Le message de succès de la création du processus est affiché.
- Aucun autre comportement supplémentaire n'est déclenché. Le fonctionnement reste donc identique à celui des versions précédentes.

Sélection du filtre dans la liste des tâches

Lorsque le processus est lancé depuis la boîte de tâches, le filtre sélectionné dépend d'un ordre de priorité précis :

- Filtre **Récent**, s'il est disponible.
- Si la tâche est affectée directement à l'utilisateur : sélection du filtre **Tâches commencées** (si présent).
- Si la tâche est non acceptée : sélection du filtre **Tâches groupe commun** (si présent).
- Filtre **Toutes les tâches**, s'il existe.
- Un filtre correspondant à la définition du processus, si disponible.
- Un filtre correspondant au rôle de la tâche, si disponible.
- Un filtre correspondant au nom de la tâche, si disponible.

Module Circuits de signature

Paramétrage du circuit

Gérer les modèles de signature

Depuis la version 8.5.11266, la fenêtre **Gérer les modèles de signature** a été enrichie afin d'améliorer la lisibilité et de simplifier l'administration des circuits de signature.

Ces évolutions visent à rendre la gestion des modèles plus intuitive et plus efficace au quotidien.

- Renommer un modèle

  Une nouvelle action **Renommer** permet désormais de modifier facilement le nom d'un modèle de signature existant. Cette fonctionnalité facilite l'organisation des modèles, notamment lorsque plusieurs variantes coexistent ou lorsqu'un modèle doit être clarifié pour mieux refléter son usage.

- Modifier un modèle

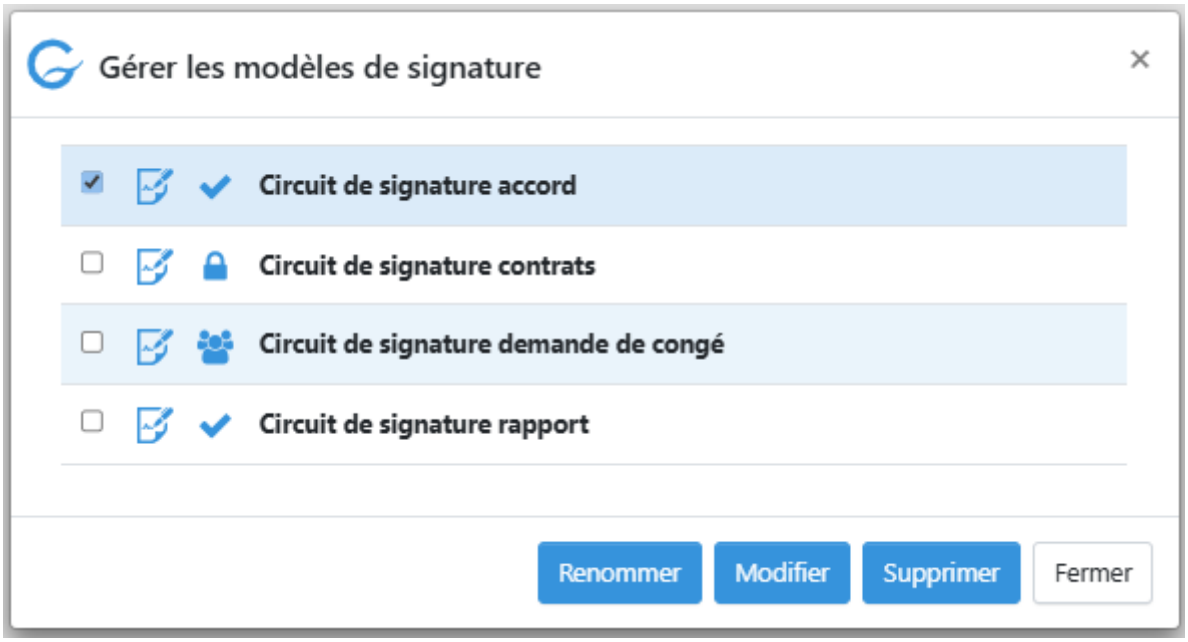
  L'action **Modifier** offre la possibilité d'ouvrir directement un modèle afin d'en ajuster le contenu ou les paramètres. Cela permet d'effectuer des mises à jour rapides sans devoir recréer un modèle ou naviguer dans des menus complexes.

- Icônes de type de circuit

  Chaque modèle de signature affiche désormais une icône dédiée indiquant le type de circuit auquel il appartient :

  - Public
  - Privé
  - Accès restreint

Cette visualisation immédiate permet d'identifier en un coup d'œil la nature du circuit et d'éviter toute confusion lors de la sélection ou de la modification d'un modèle.



Bénéfices pour l'utilisateur

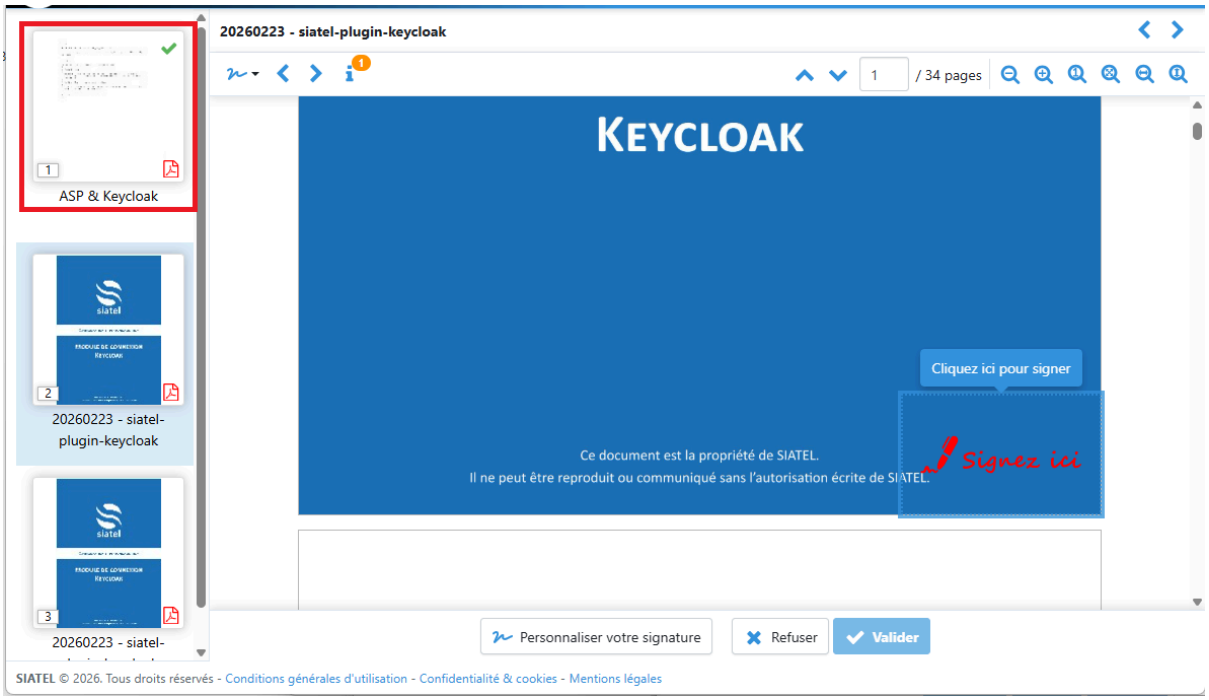
Ces améliorations apportent davantage de souplesse dans la gestion des modèles de signature et contribuent à une maintenance plus simple et plus rapide. Les administrateurs peuvent ainsi organiser, ajuster et identifier leurs modèles avec plus de précision, tout en réduisant les risques d'erreur.

SIGNATAIRES EXTERNES

MARQUAGE D'UN DOCUMENT DÉJÀ SIGNÉ

Depuis la version 8.5.11267, les documents signés par des utilisateurs externes sont désormais facilement identifiables dans l'interface. Une icône verte apparaît automatiquement sur la vignette du document une fois la signature effectuée, permettant de repérer d'un simple coup d'œil les documents signés.

Cette amélioration renforce la lisibilité du suivi des signatures.



Console d'administration

Gestion des utilisateurs

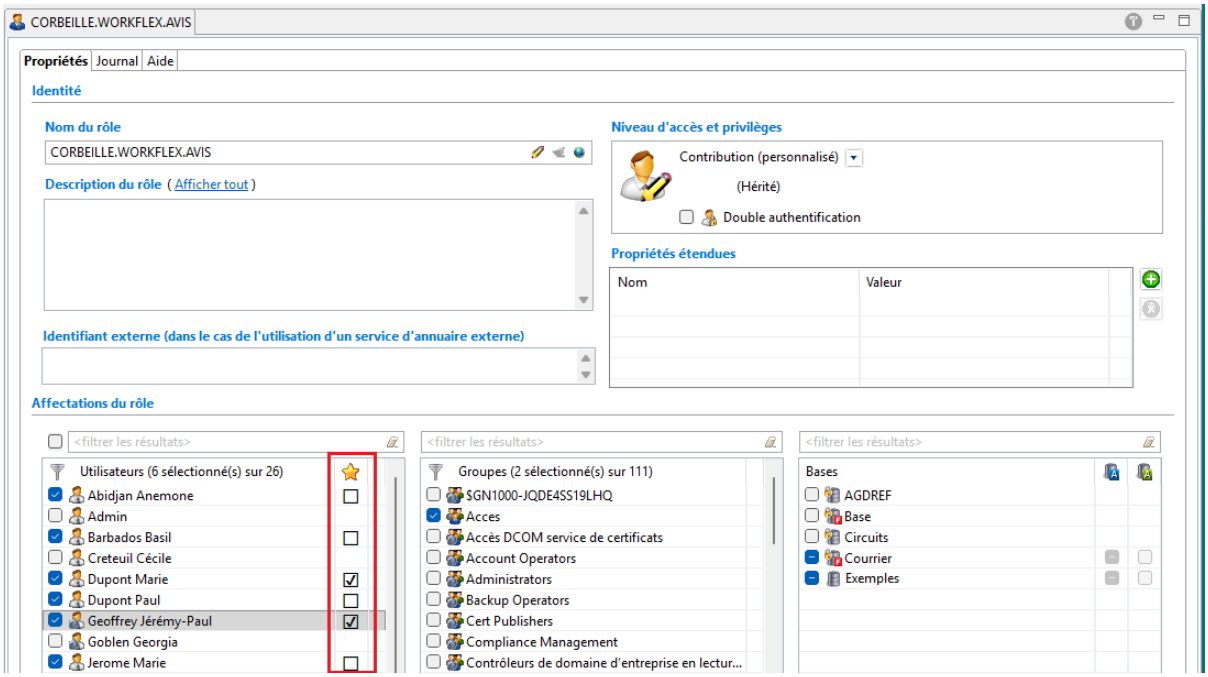
Définir un utilisateur titulaire pour un rôle fonctionnel

Depuis la version 8.5.11277, il est désormais possible de désigner un utilisateur affecté à un rôle fonctionnel comme **titulaire**.

Nouvelle colonne Utilisateur titulaire

Le tableau des affectations de rôles s'enrichit d'une nouvelle colonne intitulée « Utilisateur titulaire », contenant une case à cocher.

- Lorsque cette case est activée, l'utilisateur concerné est marqué comme titulaire pour le rôle fonctionnel associé.
- Cette indication permet de distinguer clairement l'utilisateur principal d'un rôle, notamment dans les contextes où plusieurs personnes peuvent être affectées au même rôle.



Fonctionnalité anticipant de futures évolutions

Cette nouveauté constitue une première étape vers des fonctionnalités plus avancées.

Elle est introduite en prévision de développements ultérieurs qui exploiteront la notion de titulaire pour améliorer la gestion des rôles, des responsabilités et des circuits décisionnels.

GESTION DES RÔLES

UTILISATEUR TITULAIRE POUR UN RÔLE FONCTIONNEL

Depuis la version 8.5.11277, il est désormais possible de désigner un ou plusieurs utilisateurs comme “titulaires” d’un rôle fonctionnel.

Nouvelle colonne Utilisateur titulaire

Le tableau des affectations d’utilisateurs s’enrichit d’une nouvelle colonne intitulée **Utilisateur titulaire**, contenant une case à cocher.

Lorsque cette case est activée, l’utilisateur est marqué comme titulaire pour le rôle fonctionnel concerné.

Cette désignation permet d’identifier clairement les utilisateurs principaux associés à un rôle, notamment dans les situations où plusieurs personnes peuvent partager la même fonction.

Une évolution préparant de futures fonctionnalités

L’introduction de cette notion de titulaire constitue une première étape vers des évolutions plus avancées.

Elle sera exploitée dans des développements ultérieurs afin d’améliorer la gestion des rôles, la répartition des responsabilités et les mécanismes de sélection automatique dans les processus.

GESTION DES BASES DOCUMENTAIRES

GESTION DES TYPES

GESTION DES GRILLES

NOUVELLES PROPRIÉTÉS DE MISE EN PAGE POUR LES COLONNES DE LA GRILLE

Depuis la version 8.5.11268, plusieurs nouvelles propriétés de mise en forme ont été introduites pour les colonnes des grilles, offrant un contrôle plus précis sur leur affichage et leur comportement.



Nouvelles propriétés disponibles

Visible : Oui / Non

Cette propriété indique si la colonne doit être affichée à l'écran.

Valeur par défaut : **Oui**

Pour les grilles plus anciennes ne disposant pas encore de cette propriété : si la largeur d'une colonne est définie à 0 px, la propriété Visible est automatiquement positionnée sur Non.

Exclu : Non / Oui

Cette propriété permet d'exclure totalement une colonne de l'affichage.

Une colonne marquée comme Exclue ne peut en aucun cas être rendue visible, même via un script.

Valeur par défaut : **Non**



Lorsque Exclu = Oui : les propriétés Largeur et Visible sont ignorées.

Redimensionnable : Non / Oui

Cette propriété détermine si l'utilisateur peut modifier la largeur de la colonne.

Valeur par défaut : **Non**

Si le redimensionnement est activé, il est recommandé d'activer également la barre de défilement horizontale de la grille.

Sans cela, certaines colonnes situées à droite pourraient devenir inaccessibles.

Comportement des colonnes à largeur non spécifiée

Lorsque certaines colonnes n'ont pas de largeur définie (leur taille étant calculée automatiquement en fonction de l'espace restant) :

- leur largeur s'ajuste dynamiquement lors du redimensionnement d'une autre colonne ;
- elles s'agrandissent ou se réduisent progressivement jusqu'à atteindre leur largeur minimale ;



Ce comportement est normal et attendu.

Fonctionnalités supplémentaires

Ajustement automatique de la largeur

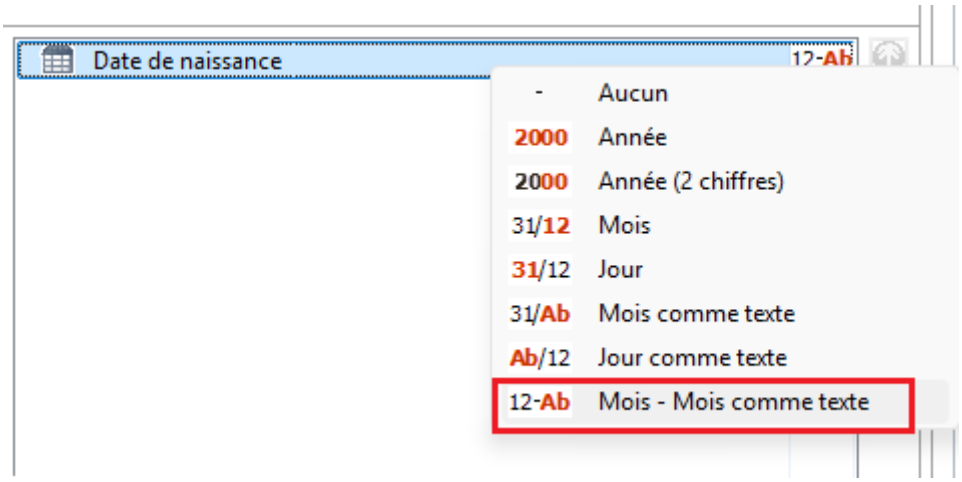
Comme dans la liste des documents, un double-clic sur l'en-tête d'une colonne ajuste automatiquement sa largeur en fonction du contenu le plus large.

- Largeur minimale : 50 px
- Largeur maximale : 1 200 px

GESTION DES RÈGLES DE COMPOSITION

NOUVEL OPÉRATEUR POUR LES CHAMPS D'INDEXATION DE TYPE DATE

Depuis la version 8.5.11272, un nouvel opérateur intitulé « Mois-Mois comme texte » a été introduit pour les champs d'indexation de type date. Cette évolution vise à harmoniser le comportement de ces champs avec celui déjà disponible dans les vues, afin d'assurer une cohérence fonctionnelle dans l'ensemble de l'application.



Objectif de l'amélioration

Jusqu'à présent, certains traitements appliqués aux dates dans les vues permettaient d'obtenir une représentation textuelle du couple mois-mois (par exemple pour générer des regroupements, des tris ou des filtres plus lisibles). Les champs d'indexation ne disposaient pas encore de cette capacité, ce qui pouvait entraîner des différences de comportement entre les vues et les index.

L'ajout de cet opérateur permet désormais :

- d'extraire la période mois-mois sous forme textuelle directement depuis un champ d'indexation de type date ;
- d'uniformiser les traitements entre les vues et les index ;
- de simplifier la configuration des filtres, regroupements ou critères basés sur des périodes mensuelles.

Un opérateur pensé pour la cohérence fonctionnelle

Grâce à cette nouveauté, les administrateurs et configurateurs bénéficient d'un outil supplémentaire pour manipuler les dates de manière homogène dans l'ensemble de la plateforme.

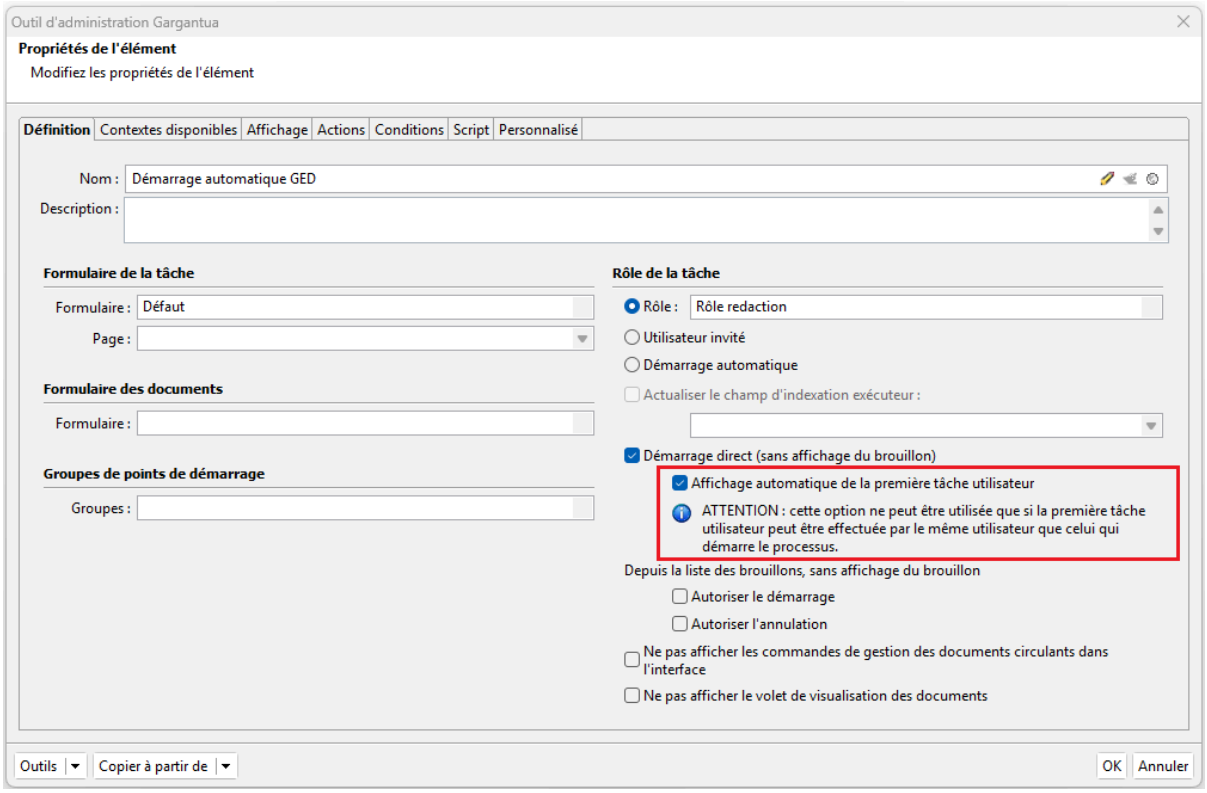
Cela facilite notamment :

- la création de critères de recherche plus lisibles,
- la mise en place de regroupements cohérents,
- la standardisation des comportements entre différents modules.

Gestion des processus

Nouvelle option pour les tâches avec démarrage direct

Depuis la version 8.5.11269, une nouvelle option a été introduite pour les tâches avec démarrage direct, permettant l'affichage automatique de la première tâche utilisateur dès le lancement du processus.



Objectif de la fonctionnalité

Cette amélioration vise à offrir une transition plus fluide entre le démarrage d'un processus et la prise en charge de la première tâche utilisateur. Elle permet à l'utilisateur d'accéder immédiatement à la tâche qui lui est destinée, sans avoir à naviguer manuellement dans la boîte de tâches.

Conditions d'application

Cette fonctionnalité s'active uniquement lorsque l'utilisateur qui lance le processus :

- est directement affecté à la tâche suivante, ou
- est affecté au rôle fonctionnel associé à cette tâche.

Dans ces cas, l'interface détecte automatiquement la première tâche utilisateur disponible et l'affiche dès qu'elle devient accessible.

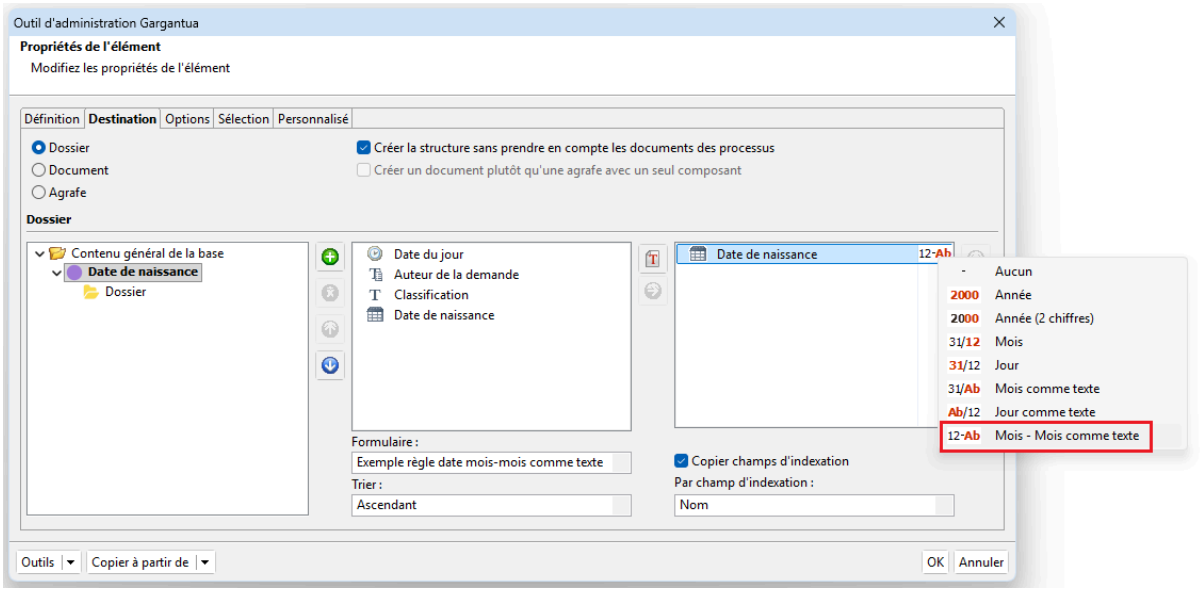
Fonctionnement général

Lorsque l'option est activée, le système surveille l'avancement du processus immédiatement après son démarrage. Dès que la première tâche utilisateur correspondant à l'utilisateur courant est créée et accessible, elle est automatiquement affichée, permettant un enchaînement rapide et naturel entre le lancement du processus et l'exécution de la tâche.

Cette logique améliore l'efficacité des utilisateurs impliqués dans des processus à démarrage direct, en réduisant les manipulations nécessaires et en rendant l'expérience plus fluide.

CLASSIFICATION GED : AJOUT DE L'OPÉRATEUR « MOIS-MOIS COMME TEXTE » POUR LES CHAMPS DE TYPE DATE

Depuis la version 8.5.11272, un nouvel opérateur intitulé « Mois-Mois comme texte » a été introduit pour les champs d'indexation de type date utilisés dans la classification des tâches GED.



Cette évolution vise à uniformiser le comportement de ces champs avec celui déjà présent dans les vues, où cet opérateur était déjà disponible. Elle permet ainsi d'obtenir une cohérence fonctionnelle dans l'ensemble du système, en particulier lors de la manipulation de dates dans les mécanismes de tri, de regroupement ou de filtrage.

Pourquoi cet opérateur ?

Dans les vues, il était déjà possible d'extraire une représentation textuelle du couple mois-mois (par exemple « 01-02 » ou « janvier-février ») afin de faciliter la lecture et l'exploitation des données temporelles.

Les champs d'indexation utilisés dans la classification GED ne disposaient pas encore de cette capacité, ce qui pouvait entraîner des différences de comportement entre les vues et les index.

L'ajout de cet opérateur permet désormais :

- d'extraire automatiquement la période mois-mois sous forme textuelle à partir d'un champ date,
- d'harmoniser les traitements entre les vues et la classification GED,
- de simplifier la configuration des critères de classement basés sur des périodes mensuelles,
- d'assurer une cohérence d'affichage et de logique métier dans l'ensemble de la plateforme.

Un pas vers une meilleure cohérence globale

Grâce à cette nouveauté, les administrateurs et configurateurs disposent d'un outil supplémentaire pour manipuler les dates de manière homogène, que ce soit dans les vues, les index ou les mécanismes de classification GED.

Cela contribue à une expérience plus prévisible, plus intuitive et plus simple à maintenir.

Gestion des scripts

Exécution d'un script sur la version brouillon

Depuis la version 8.5.11265, il est désormais possible d'exécuter un script directement depuis sa version brouillon, sans avoir à le publier au préalable.

Cette évolution apporte un gain de temps significatif pour les administrateurs et les développeurs, en leur permettant de tester immédiatement leurs modifications, d'ajuster le comportement du script et de valider son fonctionnement avant toute mise en production.

Objectif de la fonctionnalité

Jusqu'à présent, il était nécessaire de publier un script pour pouvoir l'exécuter, ce qui impliquait plusieurs étapes et pouvait ralentir les phases de test ou de mise au point.

Avec cette nouveauté :

- les cycles de développement sont plus rapides,
- les ajustements successifs sont plus fluides,
- les risques d'erreurs en production sont réduits, car le script peut être validé en amont dans son état de brouillon.

Un fonctionnement pensé pour la flexibilité

L'exécution depuis le brouillon permet de :

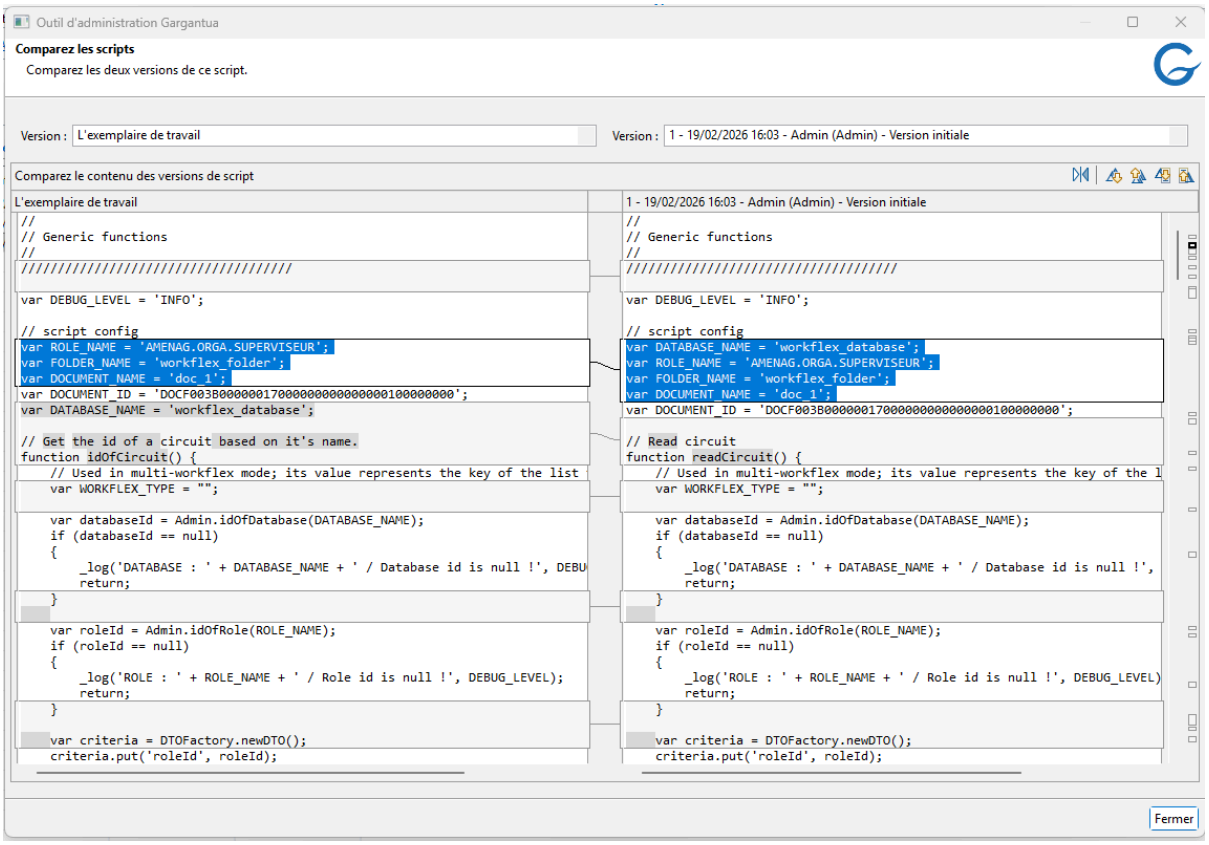
- vérifier le comportement du script dans des conditions réelles,
- tester des scénarios spécifiques sans impacter les utilisateurs,
- affiner progressivement la logique avant publication.

Cette approche facilite le travail d'expérimentation et de correction, tout en conservant la séparation claire entre les versions de test et les versions publiées.

COMPARAISON DES DEUX VERSIONS D'UN SCRIPT

Depuis la version 8.5.11280, il est désormais possible de comparer deux versions d'un même script directement depuis l'interface.

Cette fonctionnalité facilite l'analyse des évolutions apportées à un script au fil du temps et permet d'identifier rapidement les modifications effectuées.



Nouvelle interface de comparaison des versions

Dans la liste des versions d'un script, un bouton **Comparer** est désormais disponible.

En cliquant dessus, une fenêtre dédiée s'ouvre et présente deux zones distinctes, chacune permettant d'afficher une version du script.

Sélection des versions à comparer

Dans chaque zone :

- une liste déroulante permet de choisir la version du script à afficher ;
- il est possible de sélectionner n'importe quelle version, y compris la version de travail (brouillon), afin de comparer une version publiée avec une version en cours de modification.

Cette flexibilité permet d'effectuer des comparaisons précises, que ce soit pour valider un changement,

comprendre une régression ou analyser l'historique du script.

Mise en évidence des différences

Une fois les deux versions sélectionnées :

- les différences entre les scripts sont automatiquement détectées ;
- les modifications sont mises en évidence visuellement, facilitant l'identification des ajouts, suppressions ou ajustements.

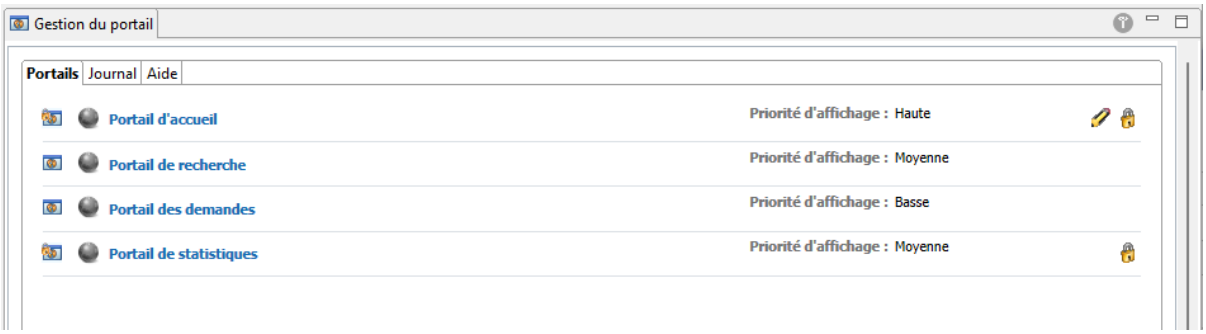
Ce mécanisme offre un véritable confort d'analyse, notamment lors de phases de maintenance, de revue ou de validation technique.

Gestion des portails

Affichage de la priorité dans la liste des portails

Depuis la version 8.5.11266, la priorité d'affichage des portails est désormais visible directement dans la liste des portails.

Cette amélioration rend la gestion de l'ordre d'apparition des portails beaucoup plus intuitive et permet aux administrateurs de visualiser immédiatement la hiérarchie appliquée.



Objectif de l'amélioration

Jusqu'à présent, la priorité d'affichage n'était pas directement accessible depuis la liste, ce qui pouvait rendre la configuration moins lisible, notamment lorsque plusieurs portails coexistaient avec des niveaux de priorité différents.

L'ajout de cette information dans la liste permet :

- d'identifier rapidement l'ordre d'apparition des portails,
- de simplifier l'organisation et la maintenance de la configuration,
- de réduire les risques d'erreur lors de la modification des priorités.

Un affichage plus clair et plus opérationnel

La priorité est désormais affichée sous forme de colonne ou d'indicateur visuel (selon la configuration), ce qui permet :

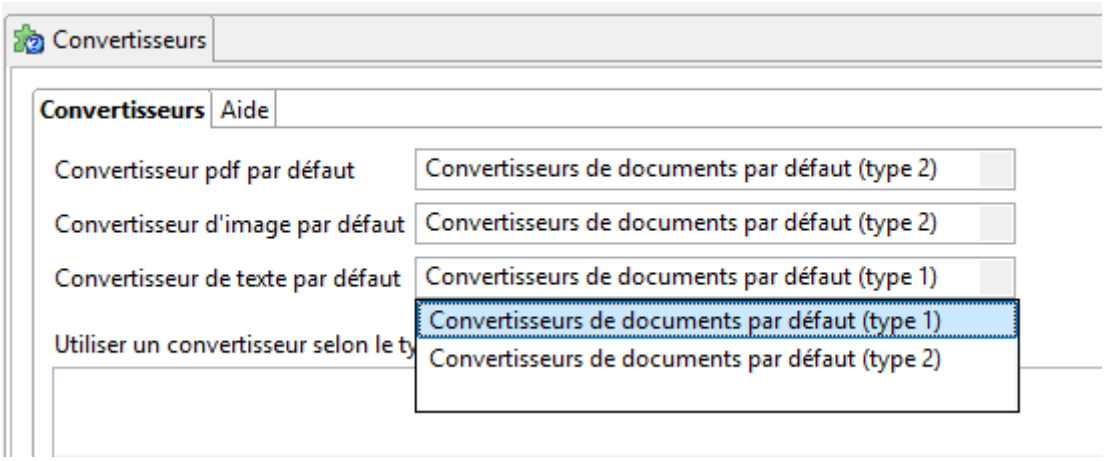
- de comparer facilement les priorités entre portails,
- de repérer immédiatement les portails prioritaires,
- d'ajuster l'ordre d'apparition de manière plus efficace.

Cette évolution contribue à une meilleure lisibilité globale de la configuration et facilite le travail des administrateurs, en particulier dans les environnements comportant un grand nombre de portails.

CONVERTISSEURS

PERSONNALISATION DU CONVERTISSEUR D'USAGE GÉNÉRAL PAR DÉFAUT

Depuis la version 8.5.11568, il est désormais possible de choisir les convertisseurs d'usage général par défaut, offrant ainsi une plus grande flexibilité dans la configuration des traitements de conversion.



Contexte avant la mise à jour

Jusqu'à présent, lorsque les deux modules de convertisseurs par défaut - type 1 et type 2 - étaient installés et activés simultanément :

- les convertisseurs de type 2 étaient automatiquement sélectionnés comme convertisseurs par défaut ;
- il n'existait aucun moyen de modifier ce comportement, même si un autre choix était souhaité pour des raisons techniques, de compatibilité ou de performance.

Cette limitation pouvait entraîner des contraintes dans certains environnements où l'utilisation des convertisseurs de type 1 restait préférable.

Nouveauté apportée

Avec cette version, bien que les convertisseurs de type 2 restent sélectionnés par défaut pour assurer la

continuité du fonctionnement existant, l'utilisateur dispose désormais de la possibilité de :

- modifier le choix du convertisseur par défaut,
- sélectionner explicitement les convertisseurs de type 1 ou de type 2,
- adapter la configuration en fonction des besoins spécifiques de l'organisation ou des exigences techniques des documents traités.

Bénéfices de cette évolution

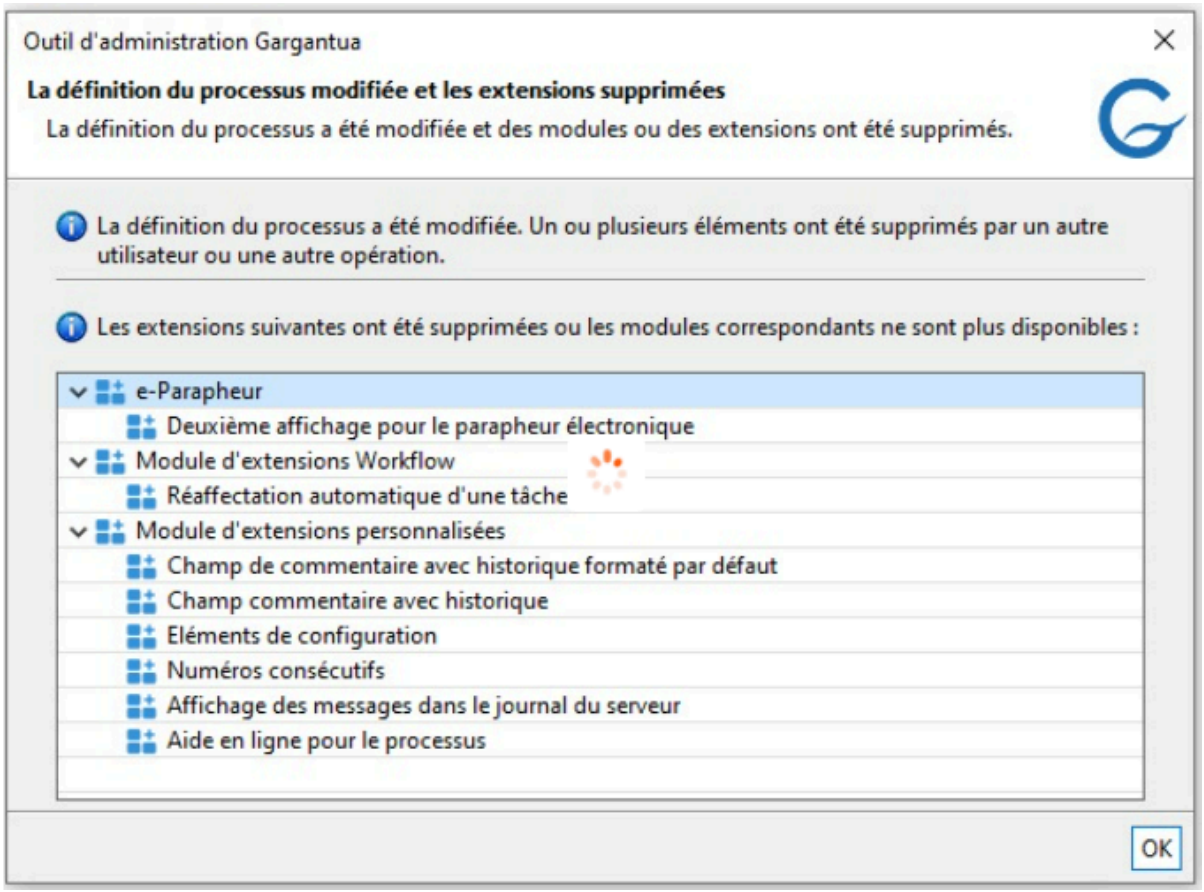
Cette amélioration apporte :

- une plus grande maîtrise de la configuration des convertisseurs ;
- une meilleure compatibilité avec des environnements hétérogènes ;
- la possibilité d'optimiser les performances ou la qualité de conversion selon les cas d'usage ;
- une réduction des contraintes liées au comportement imposé par défaut dans les versions précédentes.

GÉNÉRALITÉS

INDICATION DES EXTENSIONS MANQUANTES EN CAS D'INDISPONIBILITÉ DU MODULE

Depuis la version 8.5.11265, lorsque qu'une extension devient indisponible en raison d'un module manquant, l'interface affiche désormais automatiquement une fenêtre d'information permettant d'identifier précisément l'origine du problème.



Objectif de l'amélioration

Auparavant, lorsqu'une extension ne pouvait pas être utilisée faute de module requis, l'utilisateur ne disposait que d'indications limitées, ce qui pouvait rendre le diagnostic difficile.

Cette nouvelle fenêtre vise à :

- clarifier immédiatement la cause de l'indisponibilité,
- indiquer les modules manquants,
- lister les extensions impactées,
- faciliter la résolution rapide du problème par les administrateurs.

Une fenêtre d'information plus explicite

Lorsque la situation se produit :

- une fenêtre s'ouvre automatiquement,
- elle affiche les modules requis mais absents,
- ainsi que les extensions dépendantes qui ne peuvent pas fonctionner sans ces modules.

Cette présentation permet de comprendre en un coup d'œil ce qui doit être installé ou activé pour rétablir le fonctionnement normal.

Bénéfices pour les administrateurs

Grâce à cette évolution, la gestion des extensions devient plus simple et plus transparente :

- diagnostic plus rapide,
- réduction des erreurs de configuration,
- meilleure visibilité sur les dépendances entre modules et extensions,
- amélioration de la maintenance globale du système.



Les modules et extensions manquants sont également enregistrés dans le journal de l'outil d'administration, affichés sous forme d'arborescence avec un niveau d'information INFO.

Outil de gestion les traductions

Depuis la version 8.5.11266, un nouvel outil centralisé de gestion des traductions est disponible dans l'application.

Cet outil a été conçu pour simplifier et unifier la gestion de l'ensemble des traductions, qu'elles concernent les éléments techniques de la base ou les objets transverses du serveur.

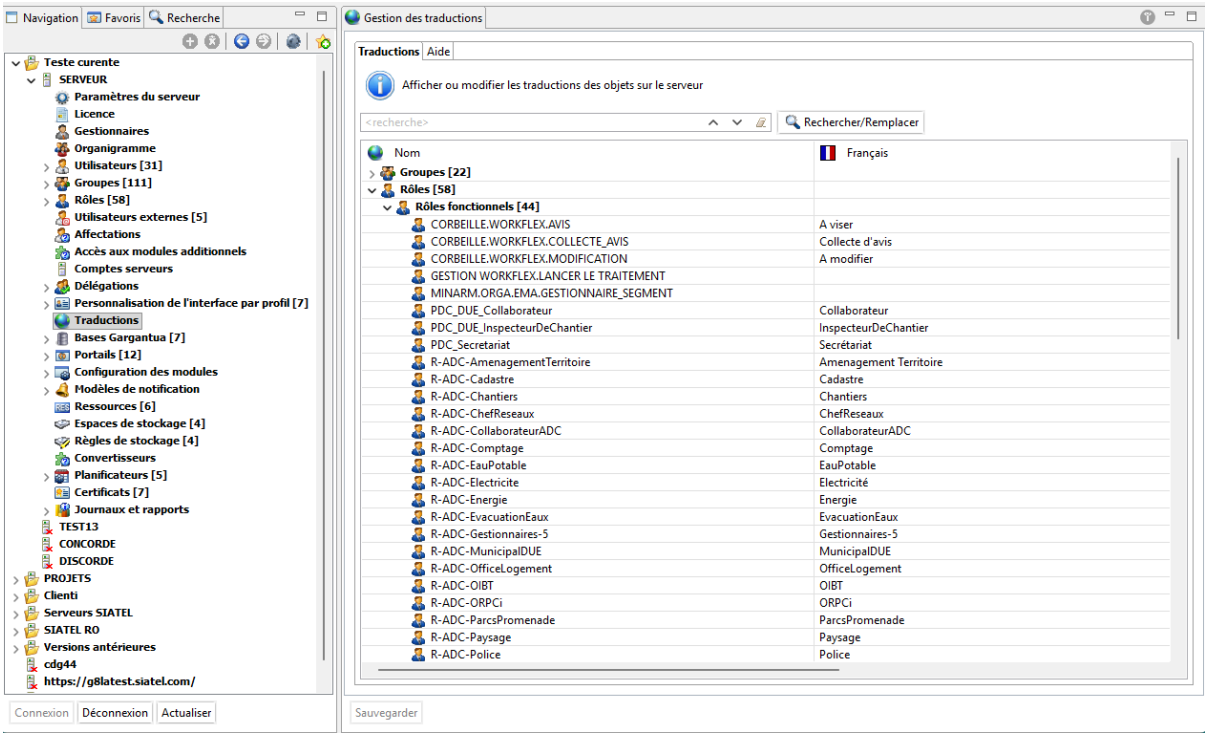
Deux niveaux de gestion des traductions

L'outil propose deux modes d'accès distincts, chacun couvrant un périmètre spécifique :

- les groupes,
- les rôles,
- les portails.

- les champs d'indexation,
- les formulaires,
- les processus,
- et plus largement tous les objets configurés dans la base.

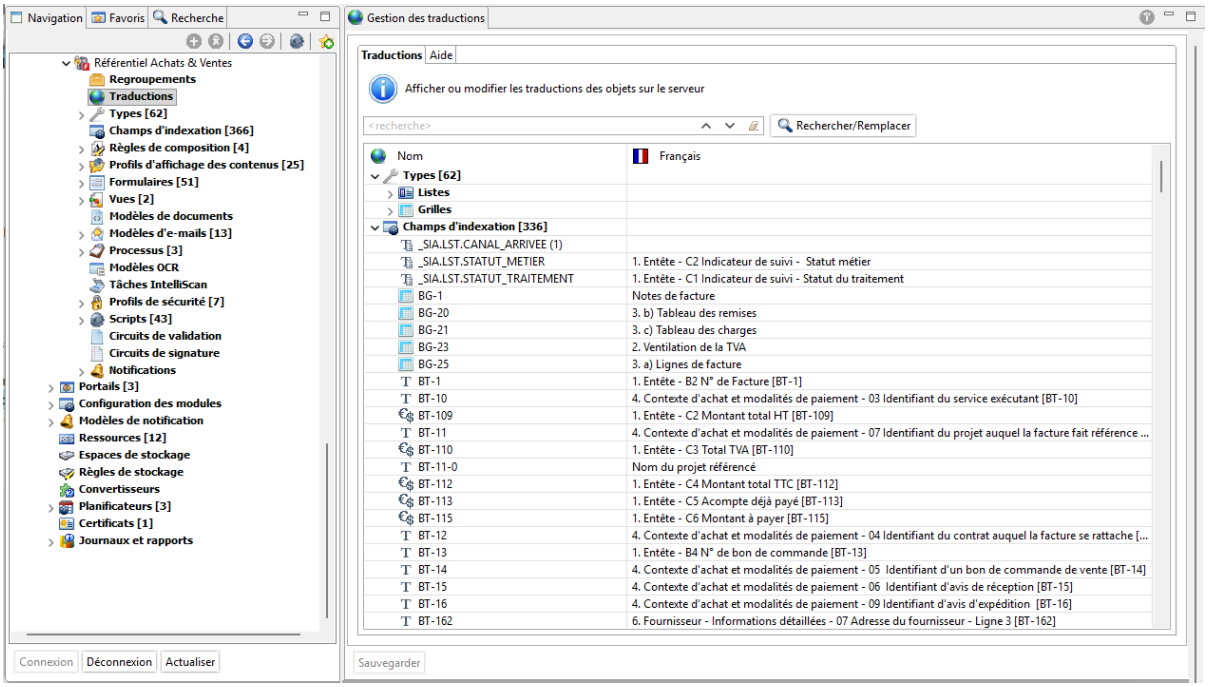
Disponible par serveur



Ce mode permet de gérer les traductions liées aux éléments transverses du serveur, notamment :

Il facilite la maintenance des traductions globales, indépendamment des bases de données.

Disponible par base



Ce mode est dédié aux traductions propres à une base donnée, incluant :

Ce découpage permet une gestion fine et adaptée à chaque contexte d'utilisation.

Une centralisation pensée pour l'efficacité

La mise à disposition d'un outil unique pour gérer toutes les traductions permet :

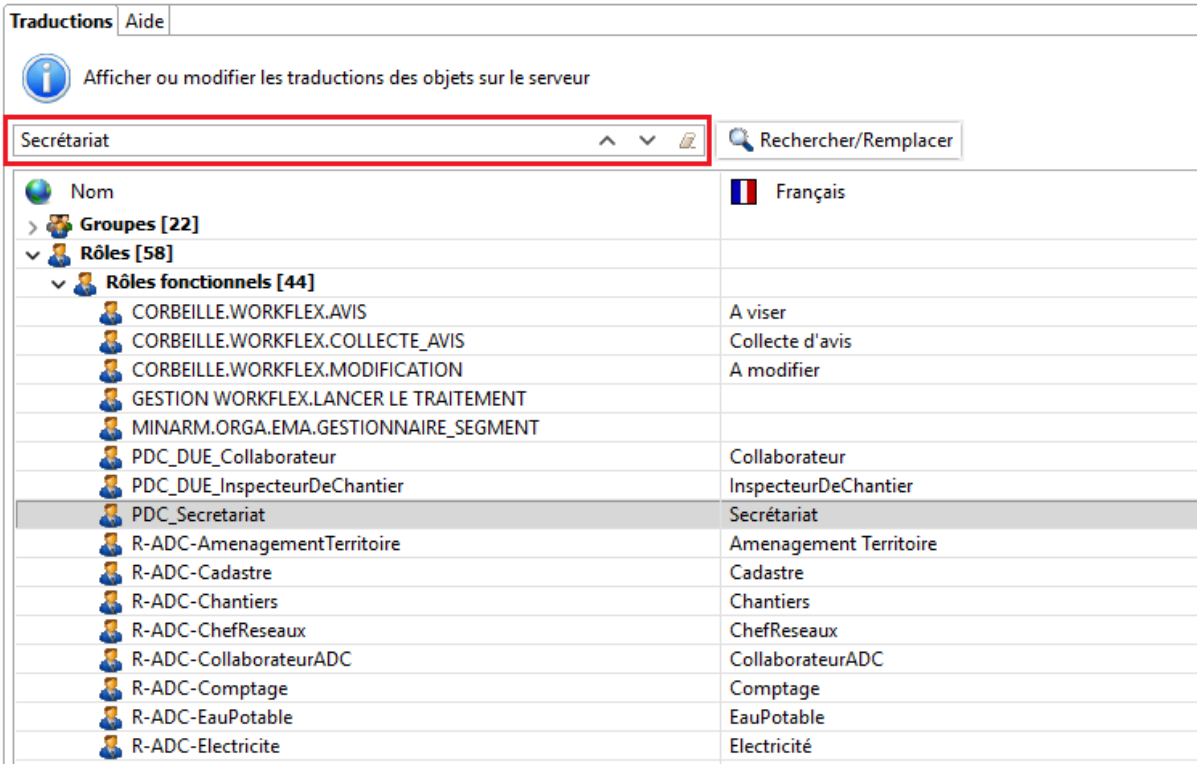
- un gain de temps significatif,
- une réduction des erreurs liées à la dispersion des points de configuration,
- une cohérence linguistique renforcée dans l'ensemble de l'application,
- une vision globale des traductions existantes et de leurs variantes.

Fonctions avancées de recherche et remplacement

Pour faciliter la maintenance des traductions, l'outil intègre plusieurs fonctionnalités puissantes :

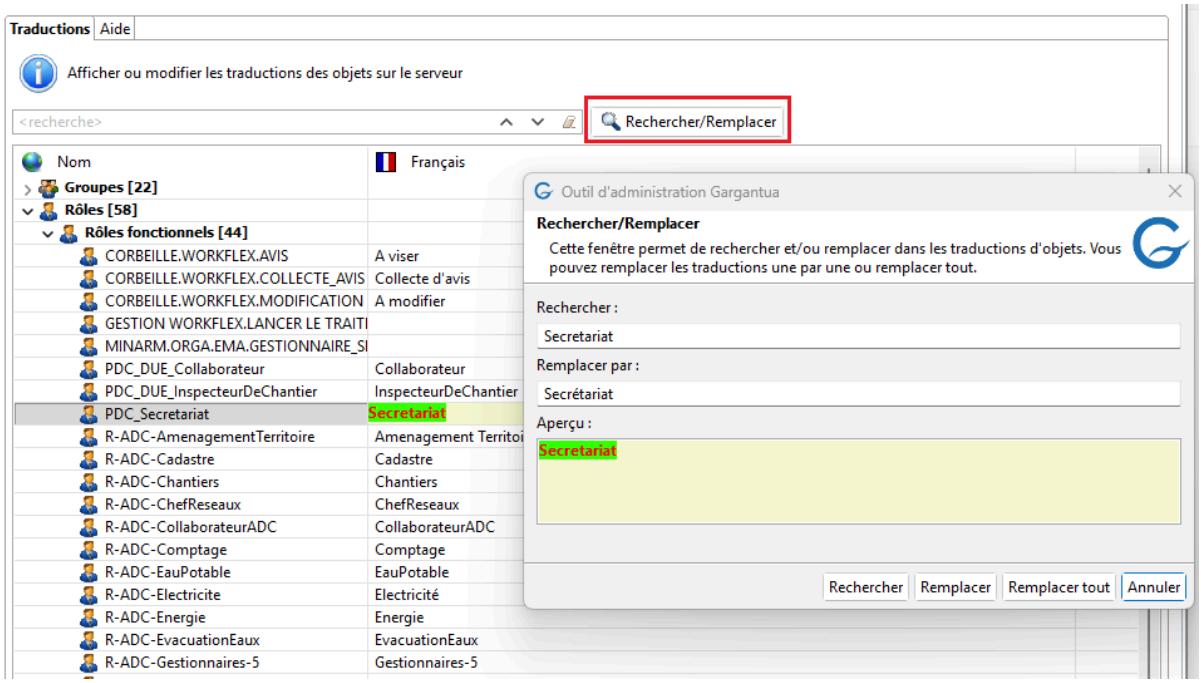
- d'effectuer des recherches ciblées sur les traductions,
- d'appliquer rapidement des remplacements,
- de corriger ou harmoniser des termes sur l'ensemble de l'application.

Recherche par nom ou par traduction



L'utilisateur peut rechercher une entrée soit par son identifiant, soit par le texte traduit, ce qui accélère la localisation des éléments à modifier.

Boîte de dialogue Rechercher / Remplacer



Un bouton dédié ouvre une fenêtre permettant :

Affichage détaillé des traductions

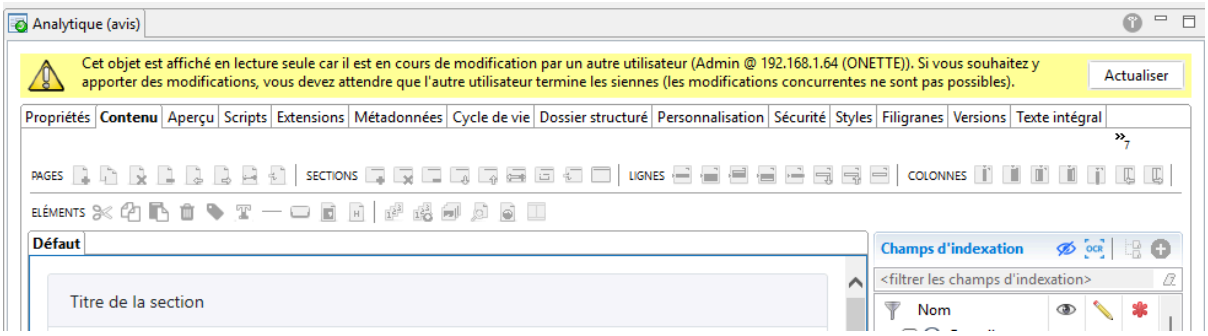
En double-cliquant sur un nom, une boîte de dialogue s'ouvre pour afficher et modifier les traductions associées.

Cette vue détaillée est particulièrement utile pour les textes longs ou les traductions complexes.

Cette nouvelle interface centralisée constitue une avancée majeure pour la gestion multilingue de l'application, en rendant les opérations plus rapides, plus fiables et plus cohérentes.

Informations complémentaires dans le bandeau notifiant la lecture seule

Depuis la version 8.5.11270, le bandeau affiché lorsqu'un objet d'administration est ouvert en lecture seule a été enrichi avec de nouvelles informations destinées à faciliter le diagnostic et la compréhension de la situation.



Informations supplémentaires affichées dans le bandeau

Désormais, lorsque l'application détecte qu'un objet d'administration ne peut pas être modifié parce qu'il est verrouillé par un autre utilisateur ou en cours d'utilisation, le bandeau d'avertissement affiche :

- l'adresse IP de la machine ayant verrouillé l'objet,
- le nom de l'ordinateur, lorsque celui-ci peut être identifié.

Ces informations viennent compléter les données déjà présentes, permettant d'identifier plus précisément l'origine du verrouillage.

Objectif de l'amélioration

Cette évolution vise à :

- faciliter le diagnostic en cas de verrouillage inattendu d'un objet d'administration,
- aider les administrateurs à identifier rapidement l'utilisateur ou la machine à l'origine du verrouillage,
- réduire les temps d'investigation, notamment dans les environnements multi-utilisateurs ou distribués,
- améliorer la transparence et la traçabilité des actions d'administration.

Un apport utile pour la maintenance

Grâce à ces informations supplémentaires, les administrateurs disposent d'une meilleure visibilité sur les situations de lecture seule, ce qui leur permet :

- de contacter plus facilement la personne concernée,
- de vérifier si un poste est resté verrouillé par erreur,
- d'éviter les blocages prolongés lors de la configuration ou de la maintenance.

Modules complémentaires et autres outils

N° de version des modules complémentaires

Extension Google Chrome (Module complémentaire spécifique à Google Chrome)

8.9702

Extension Firefox (Module complémentaire spécifique à Mozilla Firefox)

8.9703

Module complémentaire pour tous les navigateurs

8.5.11264

Modules complémentaires Microsoft Office

8.5.11264

Modules complémentaires Microsoft Outlook

8.5.11264

Modules complémentaires Mozilla Thunderbird

8.5.11264

Siatel OpenOffice extension

8.5.11264

Siatel LibreOffice extension

8.5.11264

Siatel Zimbra zimlet

8.5.11264

N° des versions des autres outils

Versions des outils disponibles

Intelliscan ver. 8.4.11264

20260311 - R8.5.11264 - IS

EasyScan ver. 8.4.11264

20260311 - R8.5.11264 - EasyScan

Custom Scan [DOSSIERS RH] ver. 8.4.11264

20260311 - R8.5.11264 - Custom Scan [DOSSIERS RH]

Media Tool ver. 8.4.11264

20260311 - R8.5.11264 - Siatel Media Tool

Media Tool Redist ver. 8.4.11264

20260311 - R8.5.11264 - Siatel Media Tool Redist

MISE À JOUR DU MICROSOFT EDGE WEBVIEW2 VERS LA DERNIÈRE VERSION

Depuis la version 8.5.11264 des modules complémentaires, des outils bureautiques et des autres composants GARGANTUA, plusieurs mises à jour techniques importantes ont été intégrées afin d'améliorer la stabilité, la compatibilité et les performances globales de l'environnement.

Mise à jour du runtime Microsoft Edge WebView2

Le runtime Microsoft Edge WebView2 a été mis à jour vers la version 1.0.3800.47.

Cette évolution garantit :

- une meilleure compatibilité avec les versions récentes de Microsoft Edge,
- des performances renforcées lors de l'affichage des interfaces intégrées,
- une sécurité accrue grâce aux correctifs inclus dans cette version,
- une expérience plus fluide pour les utilisateurs des modules reposant sur WebView2.

Cette mise à jour contribue à maintenir l'écosystème GARGANTUA aligné avec les standards technologiques actuels.

Mise à jour de la bibliothèque Microsoft.Windows.ImplementationLibrary

En parallèle, la bibliothèque Microsoft.Windows.ImplementationLibrary a été mise à jour vers la version 1.0.210204.

Cette bibliothèque, utilisée par plusieurs composants internes, apporte :

- des optimisations de performance,
- une meilleure gestion de la mémoire,
- une fiabilité accrue dans les opérations bas niveau,
- une compatibilité renforcée avec les environnements Windows récents.

Objectif global de ces mises à jour

Ces évolutions techniques visent à :

- assurer la pérennité de la plateforme,
- améliorer la stabilité des modules complémentaires et outils bureautiques,
- garantir une compatibilité continue avec les technologies Microsoft,

- offrir une base plus robuste pour les développements futurs.

Extensions/Modules serveur

Module Workflex

Bibliothèque de règles

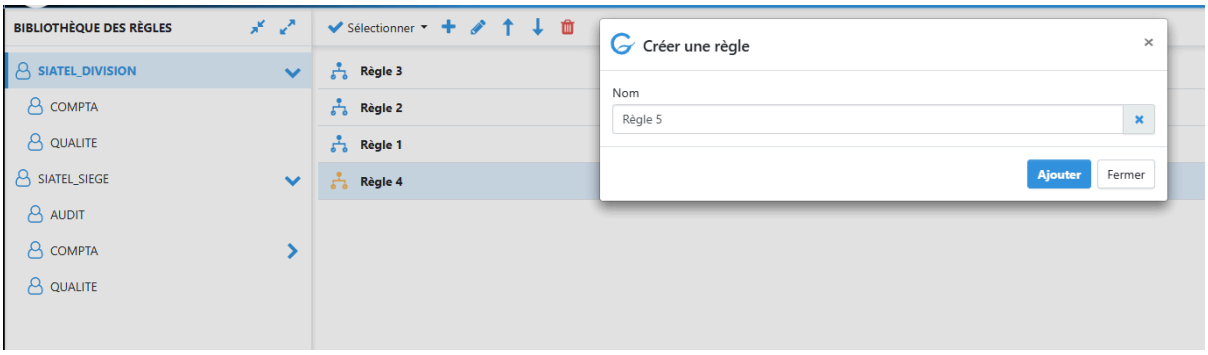
Depuis la version 8.5.11264, un nouveau composant intitulé « Bibliothèque de règles » a été ajouté au module Workflex.



Ce composant introduit un mécanisme centralisé et flexible permettant de définir, organiser et évaluer des règles personnalisées pour chaque rôle, afin d'adapter finement les circuits et comportements du module.

Création et gestion des règles par rôle

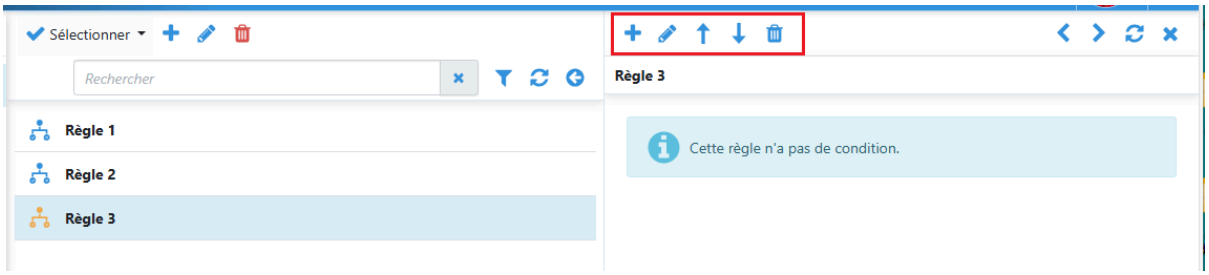
Pour chaque rôle, il est désormais possible de créer autant de règles que nécessaire.



Lorsqu'une règle est créée :

- elle ne contient initialement aucune condition,

- les conditions peuvent être ajoutées en ouvrant la règle,
- les commandes de la barre d'outils du cadre d'affichage permettent d'ajouter, modifier ou supprimer les conditions.



Cette approche offre une grande souplesse dans la définition des circuits applicables selon les situations.

Indicateurs visuels : comprendre l'état d'une règle d'un coup d'œil

La couleur de l'icône associée à chaque règle permet d'identifier rapidement son état :

- Bleu : la règle contient au moins une condition,
- Orange : aucune condition n'est définie.

Cet indicateur visuel facilite la lecture et la maintenance des règles, en particulier lorsque leur nombre devient important.

Gestion des conditions : logique d'évaluation

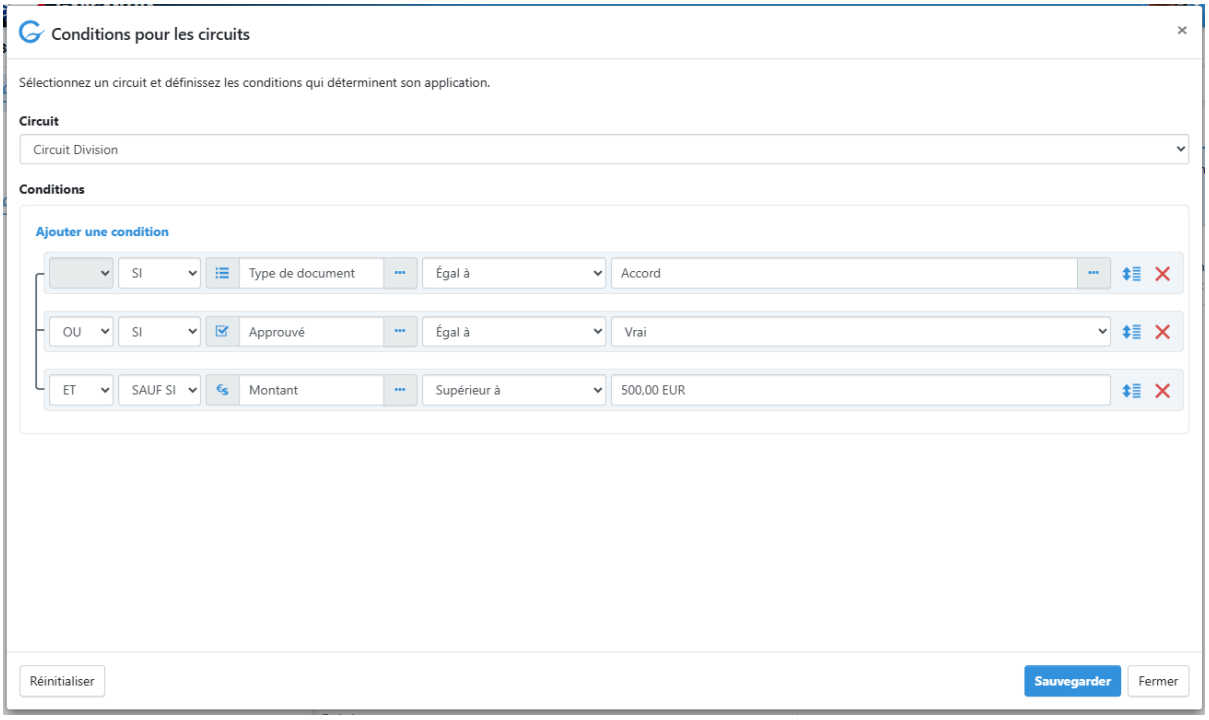
Chaque règle peut contenir un nombre illimité de conditions.

Les conditions sont évaluées dans l'ordre où elles apparaissent dans la liste :

- la première condition remplie détermine le circuit applicable,
- l'ordre des conditions peut être modifié grâce aux boutons **Déplacer vers le haut** et **Déplacer vers le bas**.

Cette logique séquentielle permet de construire des comportements précis et hiérarchisés.

Évaluation des règles : priorité et enchaînement



Les règles elles-mêmes sont également évaluées dans l'ordre de la liste :

- la première règle dont les conditions sont satisfaites est appliquée,
- les règles suivantes ne sont pas évaluées,
- l'ordre des règles peut être ajusté via les mêmes commandes de déplacement.

Ce fonctionnement garantit un contrôle total sur la priorité des règles et sur la logique de sélection du circuit final.

Ce nouveau composant renforce considérablement la puissance et la flexibilité du module Workflex, en permettant de définir des comportements complexes tout en conservant une interface claire, structurée et facile à maintenir.

Exemple de script pour évaluation des règles

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

var DEBUG_LEVEL = 'INFO';


// script config
```

```
var DATABASE_NAME = 'workflex_database';
var ROLE_NAME = 'AMENAG.ORGA.SUPERVISEUR';
var FOLDER_NAME = 'workflex_folder';
var DOCUMENT_NAME = 'doc_1';
var DOCUMENT_ID = 'DOCF003B0000001700000000000000100000000';

// Lire un circuit
function readCircuit() {
   // Utilisé dans le mode multi-workflex; La valeur représente la clé de la liste multi-workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
     _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
     return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
     _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
     return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var circuitName = 'workflex_circuit';
   var circuit = Workflex.readCircuit(circuitName, criteria);

   return circuit != null ? "Circuit DTO to string : " + circuit.toString() : null;
}
```

```
// Get the id of a circuit based on it's name.
function idOfCircuit() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
      _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
      return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
      _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
      return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var circuitName = 'workflex_circuit';
   var circuitId = Workflex.idOfCircuit(circuitName, criteria);

   _log('Circuit id : ' + circuitId, DEBUG_LEVEL);

   return "Circuit id : " + circuitId;
}

// Enumerate all circuits.
function enumerateCircuits() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";
```

```
var databaseId = Admin.idOfDatabase(DATABASE_NAME);
if (databaseId == null)
{
   _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
   return;
}

var roleId = Admin.idOfRole(ROLE_NAME);
if (roleId == null)
{
   _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
   return;
}

var criteria = DTOFactory.newDTO();
criteria.put('roleId', roleId);
criteria.put('databaseId', databaseId);
criteria.put('type', WORKFLEX_TYPE);

var circuits = new java.util.ArrayList(0);
var json = '{ "circuitList" : [ ';

try {
   Workflex.enumerateCircuits(criteria, circuits);
   if (circuits != null && !circuits.isEmpty()) {
      for (var i = 0; i < circuits.size(); i++) {
         var circuit = circuits.get(i);
        var circuitId = circuit.get("id");
        var circuitName = circuit.get("name");
         if (i > 0) {
        json = json + ', ';
        }
         json = json + '{ "circuitId" : "' + circuitId
           + '", "circuitName" : "' + circuitName + '"}';
      }
```

```
      }
   } catch (e) {
   }
   json = json + ' ] }';
   _log('Circuits json : ' + json, DEBUG_LEVEL);
   return json;
}

// Delete an existing circuit.
function deleteCircuit() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-
workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
      _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
      return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
      _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
      return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var circuitName = 'workflex_circuit';
   var circuitId = Workflex.idOfCircuit(circuitName, criteria);

   try {
```

```
      Workflex.deleteCircuit(circuitId, criteria);
    } catch (e) {
    }
}

// Read rule
function readRule() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
      _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
     return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
      _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
      return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var ruleName = 'rule_1';
   var rule = Workflex.readRule(ruleName, criteria);

   return rule != null ? "Rule DTO to string : " + rule.toString() : null;
}

// Get the id of a rule based on it's name.
```

```
function idOfRule() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
      _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
      return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
      _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
      return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var ruleName = 'rule_1';
   var ruleId = Workflex.idOfRule(ruleName, criteria);

  _log('Rule id : ' + ruleId, DEBUG_LEVEL);

   return "Rule id : " + ruleId;
}

// Enumerate all rules.
function enumerateRules() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";
```

```
var databaseId = Admin.idOfDatabase(DATABASE_NAME);
if (databaseId == null)
{
   _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
   return;
}

var roleId = Admin.idOfRole(ROLE_NAME);
if (roleId == null)
{
   _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
   return;
}

var criteria = DTOFactory.newDTO();
criteria.put('roleId', roleId);
criteria.put('databaseId', databaseId);
criteria.put('type', WORKFLEX_TYPE);

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

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

try {
    Workflex.enumerateRules(criteria, rules);
    if (rules != null && !rules.isEmpty()) {
       for (var i = 0; i < rules.size(); i++) {
          var rule = rules.get(i);
         var ruleId = rule.get("id");
         var ruleName = rule.get("name");
          if (i > 0) {
            json = json + ', ';
         }
          json = json + '{ "ruleId" : "' + ruleId
            + '", "ruleName" : "' + ruleName + '"}';
      }
   }
```

```
    } catch (e) {
    }
    json = json + ' ] }';
    _log('Rules json : ' + json, DEBUG_LEVEL);
    return json;
}

// Delete an existing rule.
function deleteRule() {
    // Used in multi-workflex mode; its value represents the key of the list for multi-
workflex.
    var WORKFLEX_TYPE = "";

    var databaseId = Admin.idOfDatabase(DATABASE_NAME);
    if (databaseId == null)
    {
        _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
        return;
    }

    var roleId = Admin.idOfRole(ROLE_NAME);
    if (roleId == null)
    {
        _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
        return;
    }

    var criteria = DTOFactory.newDTO();
    criteria.put('roleId', roleId);
    criteria.put('databaseId', databaseId);
    criteria.put('type', WORKFLEX_TYPE);

    var ruleName = 'rule_1';
    var ruleId = Workflex.idOfRule(ruleName, criteria);

    try {
        Workflex.deleteRule(ruleId, criteria);
```

```
    } catch (e) {
    }
}

// Evaluate an object's circuits.
function evaluateObjectCircuits() {
   // Used in multi-workflex mode; its value represents the key of the list for multi-workflex.
   var WORKFLEX_TYPE = "";

   var databaseId = Admin.idOfDatabase(DATABASE_NAME);
   if (databaseId == null)
   {
      _log('DATABASE : ' + DATABASE_NAME + ' / Database id is null !', DEBUG_LEVEL);
      return;
   }

   var roleId = Admin.idOfRole(ROLE_NAME);
   if (roleId == null)
   {
      _log('ROLE : ' + ROLE_NAME + ' / Role id is null !', DEBUG_LEVEL);
      return;
   }

   var criteria = DTOFactory.newDTO();
   criteria.put('roleId', roleId);
   criteria.put('databaseId', databaseId);
   criteria.put('type', WORKFLEX_TYPE);

   var document = SiatelFactory.newObject(Prefix.FILE_DOCF);
   document.setId(DOCUMENT_ID);

   Ged.readDocument(document, _l(F.DOCUMENT_FULL));

   try {
      var dto = Workflex.evaluateObjectCircuits(document, criteria);
      var circuitId = dto != null ? dto.get("circuitId") : null
```

```
        _log('Circuit id : ' + circuitId, DEBUG_LEVEL);
    } catch (e) {
    }
}
```

Autres développements

API scripting formulaires

Support pour télécharger des documents depuis un formulaire

Depuis la version 8.5.11264, une nouvelle fonctionnalité a été introduite permettant de télécharger des documents directement depuis un formulaire.

Cette évolution enrichit les capacités d'interaction avec les documents au sein des formulaires et simplifie considérablement les actions réalisées par les utilisateurs ou les scripts.

Nouvelle méthode : form_downloadDocuments(ids)

Une nouvelle méthode a été ajoutée :

```
form_downloadDocuments(ids)
```

Paramètre

ids

un tableau d'identifiants de documents à télécharger.

Chaque identifiant correspond à un document existant dans la base.

Cette méthode peut être appelée depuis un script associé à un formulaire, permettant ainsi de déclencher le téléchargement d'un ou plusieurs documents en une seule action.

Objectif de la fonctionnalité

Avant cette version, le téléchargement de documents depuis un formulaire nécessitait des manipulations plus complexes ou des actions manuelles.

Grâce à cette nouvelle méthode :

- les scripts peuvent initier directement le téléchargement,
- les utilisateurs bénéficient d'une interaction plus fluide,
- les processus intégrant des documents deviennent plus simples à automatiser.

Cas d'usage typiques

- Bouton dans un formulaire permettant de télécharger automatiquement un ensemble de documents liés.
- Script déclenchant le téléchargement d'un document généré ou sélectionné.

- Automatisation de la récupération de pièces jointes dans un processus métier.

SUPPORT POUR AFFICHER/CACHER LES COLONNES D'UNE GRILLE

Depuis la version 8.5.11268, une nouvelle méthode de contrôle a été ajoutée pour les grilles :

Nouvelle méthode : toggleHiddenColumns()

Cette méthode permet d'afficher ou de masquer dynamiquement les colonnes d'une grille, offrant ainsi un moyen simple et rapide de gérer la visibilité des colonnes directement depuis un script associé à un formulaire.

Objectif de la fonctionnalité

Jusqu'à présent, la gestion de la visibilité des colonnes nécessitait des manipulations plus complexes ou des configurations statiques.

Avec cette nouvelle méthode :

- l'utilisateur peut alterner l'affichage des colonnes masquées,
- les scripts peuvent adapter l'affichage en fonction du contexte,
- l'interface devient plus flexible et plus ergonomique.

Cette fonctionnalité est particulièrement utile dans les formulaires comportant de nombreuses colonnes ou lorsque certaines colonnes ne doivent être visibles qu'à certains moments.

Utilisation de la méthode

La méthode s'utilise via le contrôle de la grille :

```
var ctrl = form_getControl(attrId);
ctrl.toggleHiddenColumns();
```

Paramètres

attrId

correspond à l'identifiant de l'attribut associé à la grille.

L'appel de la méthode inverse l'état actuel :

- si des colonnes sont masquées, elles deviennent visibles ;
- si elles sont visibles, elles sont masquées.

Point d'attention : défilement horizontal obligatoire

Comme pour la propriété **Redimensionnable**, il est indispensable d'activer le défilement horizontal de la grille lorsque vous utilisez cette méthode.

Sans cela :

- certaines colonnes situées à droite risquent de devenir inaccessibles,
- l'utilisateur pourrait ne pas voir les colonnes réaffichées,
- l'expérience globale pourrait être perturbée.

L'activation du défilement horizontal garantit une navigation fluide et un affichage complet des colonnes.

APIs scripting serveur

Support pour cacher les champs personnalisés dans les circuits de signature

Depuis la version 8.5.11274, une nouvelle option intitulée **hidden** a été ajoutée pour les champs de signature personnalisés.

Cette évolution offre davantage de contrôle sur l'affichage de ces champs et permet d'adapter leur visibilité en fonction du contexte ou de la logique métier définie dans les scripts.

Nouvelle option : hidden

L'option hidden peut être activée ou désactivée par script, ce qui permet :

- de masquer automatiquement un champ de signature dans certaines situations,
- d'adapter dynamiquement l'interface selon le rôle de l'utilisateur, l'état du document ou tout autre critère métier,
- de simplifier l'expérience utilisateur en n'affichant que les champs réellement nécessaires.

Lorsque cette option est activée, le champ concerné est masqué dans la boîte de dialogue des champs, ce qui évite toute confusion ou surcharge visuelle.

Objectif de l'amélioration

Cette fonctionnalité répond à plusieurs besoins :

- permettre une meilleure personnalisation des formulaires,
- offrir une gestion plus fine des champs de signature,
- éviter l'affichage de champs inutiles ou non pertinents selon le contexte,
- renforcer la clarté et la lisibilité des interfaces de signature.

Utilisation typique dans un script

L'option peut être configurée directement via les API de script associées aux champs de signature.

Elle s'intègre naturellement dans les scénarios où la visibilité doit être conditionnelle.