

Gendocs

- [Insertion de variables](#)
- [Insertion de documents](#)
- [Énumération](#)
- [Comptage](#)
- [Filtres](#)
- [Récupérer une liste de valeurs](#)
- [Images](#)
- [QRCodes et codes-barre](#)
- [Utilitaires](#)

Insertion de variables

Variables simples

Insère un champ issu du contexte courant.

```
{ MERGEFIELD nom_du_champ }
```

Paramètres issus du modèle

Si des paramètres ont été passés au modèle, il est possible d'en extraire la valeur à l'aide de :

```
{ DOCVARIABLE param <nom_du_paramètre> }
```

Trames et variables d'un contexte « amont »

```
{ DOCVARIABLE champ nom_du_champ english }
```

Si la mention **english** est indiquée et que le champ est de type décimal, un formatage avec le « . » comme séparateur décimal sera utilisé.

```
{ DOCVARIABLE champ nom_du_champ altformat }
```

Si la mention **altformat** est indiquée, c'est la mise en forme incluse dans les RTF insérés qui sera utilisée (par défaut c'est la mise en forme du document).

Insertion de documents

Documents simples

```
{ DOCVARIABLE insert nom_du_document }
```

Sous-documents

```
{ DOCVARIABLE doc nom_du_document contexte identifiant unique filter=XXX }
```

ou

```
{ DOCVARIABLE doc nom_du_document contexte:référence identifiant unique }
```

Énumération

```
{ DOCVARIABLE distinct nom_du_document contexte identifiant unique distinct=champ1,champ2 filt
```

S'utilise de la même façon que les { docvariable doc } avec un paramètre obligatoire supplémentaire : `distinct=XXX`.

Cette fonction appelle le nom_du_document dans le contexte mais une fois pour chaque ensemble de valeurs uniques indiquées dans le paramètre distinct=XXX. Ainsi, pour récupérer les différentes valeurs **rôle** du copyright d'une œuvre, on indiquera :

```
{ DOCVARIABLE distinct nom_du_document copyright { mergefield oeu_idoeu } distinct=cop_role,cc
```

Le document appelé, (ne) verra (que) les champs **cop_role** et **cop_chain** ainsi qu'un champ supplémentaire nommé « distinctfilter » qui permettra de filtrer les lignes de copyright pour chaque rôle et chaque lettrage :

```
{ DOCVARIABLE doc nom_du_document copyright { mergefield oeu_idoeu } filter={ mergefield disti
```

Comptage

```
{ DOCVARIABLE count X contexte identifiant_unique filter=XXX }
```

S'utilise de la même façon que les `{ docvariable doc }` mais retourne simplement le nombre de résultats (et non les résultats eux-mêmes).

Par exemple, pour compter le nombre de lignes de copyright qui ont un rôle **A** :

```
{ DOCVARIABLE count X copyright { mergefield oeu_idoeu } filter=(cop_role='A') }
```

Ce compte peut être utilisé dans des tests :

```
{ IF { DOCVARIABLE count X copyright { mergefield oeu_idoeu } filter=(cop_role='A') } > 0 "Il y a des auteurs" "Aucun auteur" }
```

Il est possible de compter le nombre de valeurs uniques d'un champ :

```
{ DOCVARIABLE count X copyright { mergefield oeu_idoeu } filter=(cop_role='A')  
distinct=cop_chain }
```

Filtres

Pour les opérations **doc**, **distinct** et **count**, il est possible de filtrer les résultats selon n'importe quel(s) champ(s) du contexte.

Les filtres doivent respecter les syntaxes suivantes :

- aucun espace n'est autorisé au sein d'un filtre
- un critère = <champ><opérateur_de_comparaison><valeur(s)>
- si plusieurs critères sont indiqués dans un filtre, ils doivent être encadrés avec des parenthèses et séparés par « & » pour signifier « et » ou « | » pour signifier « ou ».
- plusieurs niveaux de parenthèses sont autorisés pour prioriser les et/ou.

Cas particuliers

Un champ seul (sans opérateur de comparaison ni valeur) signifie « le champ n'est pas vide »

Exemple : (cop_mechowned)

Signifie que le % **mechowned** est renseigné

Un champ seul (sans opérateur de comparaison ni valeur) précédé du symbole « ! » signifie « le champ est vide »

Exemple : (!cop_mechowned)

Signifie que le % **mechowned** n'est pas renseigné

Opérateurs de comparaison

Opérateur	Signification
=	Égal à
<>	Différent de
>	Supérieur (strictement) à
<	Inférieur (strictement) à
>=	Supérieur ou égal à
≤	Inférieur ou égal à
~	Contient (spécial)
~~	Contient (spécial) sans distinction MAJ/min

Contient (spécial)

Cet opérateur est puissant mais complexe à utiliser. Il permet de vérifier qu'un champ correspond à un motif exprimé comme une « regex » ou « regexp » ou « expression régulière » (Internet fourmille de ressources sur les regex).

Exemple

Le rôle est A ou C ou CA :

```
filter=cop_role~'^(A|C|CA)$'
```

Équivalent à

```
filter=(cop_role='A')|(cop_role='C')|(cop_role='CA')
```

Le titre commence par « JAMES BOND » :

```
filter=oeu_titre~'^JAMES BOND'
```

Exemples

Le rôle est **A** :

```
filter=cop_role='A'
```

Le rôle est **A** ou **C** :

```
filter=(cop_role='A')|(cop_role='C')
```

Le rôle est A et le lettrage est A :

```
filter=(cop_role='A')&(cop_chain='A')
```

Le (rôle est A ou C) et (le lettrage est A) :

```
filter=((cop_role='A')|(cop_role='C'))&(cop_chain='A')
```

Récupérer une liste de valeurs

```
{ DOCVARIABLE select cop_nomacteur copyright { mergefield oeu_idoeu } filter=(cop_role='A') }
```

⇒ John|Paul|Georges|Ringo

Il est possible de formater le résultat “à la française” ainsi :

```
{ DOCVARIABLE select cop_nomacteur copyright { mergefield oeu_idoeu } filter=(cop_role='A')  
virgules_et }
```

⇒ John, Paul, Georges et Ringo

Images

```
{ DOCVARIABLE image "chemin_vers_le_fichier_image" }
```

QRCodes et codes-barre

QRCodes

```
{ DOCVARIABLE qrcode "texte" ratio EC=x }
```

ratio est une valeur numérique qui permet de modifier la taille du QRcode. 1=taille normale, 0.5 = 1/2 taille, 2 = double taille (attention, pour les nombres à virgule, il faut utiliser le « . » comme les anglo-saxons).

EC=x : niveau de correction d'erreur. $0 \leq x \leq 3$ (0=pas de correction, 3=correction max)

Codes-barre

```
{ DOCVARIABLE barcode "texte" typedecode ratio }
```

typedecode peut prendre l'une des valeurs suivantes : code39 code128 code128c

ratio fonctionne comme pour QRcodes (une valeur numérique par laquelle on multipliera la taille de l'image)

Utilitaires

arrondis

```
{ DOCVARIABLE arrondis 12.34567 }
```

Retourne 12.34

arrondistriche

```
{ DOCVARIABLE arrondis 12.34567 }
```

Retourne le nombre “rond” le plus proche (ici 12.35)

virgules_et

En partant d'une chaîne contenant plusieurs textes séparés par le caractère “|” (barre verticale), construire une représentation “à la française” de cette chaîne :

```
{ DOCVARIABLE virgules_et "John|Paul|Georges|Ringo" }
```

Retourne “John, Paul, Georges et Ringo”

match

Permet de comparer une chaîne à une regex (expression régulière) et retourne “True” ou “False” si la chaîne matche avec le motif :

```
{ DOCVARIABLE match "titit0Totata" "toto" }
```

⇒ True car “toto” est bien contenu dans la chaîne. Attention aux caractères spéciaux des regex (parenthèses, ^, \$, ., *, +, [], etc.)

log

```
{ DOCVARIABLE log "message" }
```

Ajout “message” dans le journal.

newpage

```
{ DOCVARIABLE newpage }
```

Force un saut de page dans le document.

floatenlettres

```
{ DOCVARIABLE floatenlettres 1234.56 [separateur=euros] }
```

Retourne la transcription du nombre en français : mille deux cent trente quatre euros cinquante six

multimatch

```
{ DOCVARIABLE multimatch "text_à_comparer" "comparaison1" "valeur1" "comparaison2" "valeur2" }
```

Effectue plusieurs comparaisons (comme la fonction `match`) et retourne une correspondance. Dans l'exemple ci-dessous, on compare la chaîne “Les éditions du chat libre” avec deux “motifs”. Si l'un des mots **editions** ou **éditions** est trouvé, la fonction retourne “éditeur”, si le mot **production** est trouvé, la fonction retourne “producteur” et sinon, la fonction retourne “autre”. La valeur par défaut est facultative.

```
{ DOCVARIABLE multimatch "Les éditions du chat libre"
  "editions|éditions"      "éditeur"
  "production"             "producteur"
                           "autre"
}
```

Notez qu'il est possible (et recommandé) de sauter des lignes et d'aligner les comparaisons et valeurs pour que tout soit plus lisible et qu'il soit aussi plus simple d'ajouter de nouvelles comparaisons/valeurs.

L'usage recommandé est de stocker le résultat dans une variable (voir `DOCVARIABLE SET`) ainsi :

```
{ docvariable set initiale "{ docvariable multimatch { docvariable champ oeu_titre }
  "[^aeiouy]" "voyelle"
  "consonne" }" }
```

La variable **initiale** prendra la valeur “voyelle” si le titre commence par une voyelle, sinon (valeur par défaut) elle contiendra “consonne”.