

Ce tutoriel va vous indiquer comment exporter des données dans des fichiers XML

✘ Cette fonction est disponible dans la révision 4571 ou supérieure.

Cette application ressemble étrangement aux applications d'export de fichiers avec renommage du fichier et des répertoires en fonction des valeurs extraites dans le Cold et/ou du Vidéocodage manuel effectué par la suite dans le centre d'indexation.

But de l'application :

- On dispose d'un scénario de Cold ou d'une requête.
- On veut, pour chaque élément exporter dans un fichier, les données , formatées selon une charte d'export bien définie.

Besoins pour réaliser ce tutoriel :

- Un scénario de Cold existant (l'exemple sera effectué sur une table complexe , contenant des factures fournisseurs issue d'un OCR multimodèle par exemple)
- La requête correspondante (ne présentant que les factures validées)
- Un fichier XML ou CSV qui comporte le modèle d'export des données que l'on appellera fichier "template"

Tutoriel

Création du scénario Cold et de la table

Assistant Cold, première étape
Merci de choisir une référence

Référence: Ocr complet une fiche descriptive par fichier, plusieurs modèles
Action: [OK]

Assistant Cold, seconde étape
Saisissez le répertoire à scruter (merci de placer des fichiers dans le répertoire à scruter, afin de préparer la maquette)

Chemin: F:\nchp\var\spool\ezged\instance\ocr\wait\four
Action: [OK]

Assistant Cold, troisième étape
Table de destination ?

Nouvelle table: flour
Table existante: adulectactes
Action: [OK]

Assistant Cold, troisième étape (détails)
Création des champs

Champ	Type	Taille	Non Obligatoire
FFOUR_ID	Numérique	20	
FFOUR_PIECE	Texte	50	
FFOUR_DATEPIECE	Date	0	
FFOUR_MONTANTHT	Flottant	0	
FFOUR_MONTANTTVA	Flottant	0	
FFOUR_MONTANTTTC	Flottant	0	
FFOUR_FOURNISSEUR	Texte	50	
FFOUR_VALIDEE	Booléen	4	
FFOUR_EXPORTEE	Booléen	4	
FFOUR_	Numérique	50	

[Continuer l'assistant ->](#)

Cette structure de table peut bien sûr être plus complexe ou plus simple, le champs FFOUR_VALIDEE booléen est nécessaire pour la création de la requête qui permettra à la fois le blocage dans le Cold avant export, ou la bonne gestion des éléments à exporter. Il sera modifiable dans la partie centre d'indexation, par les personnes devant compléter et valider les factures.

Création des requêtes nécessaires

1. Nous allons ici créer la requête nécessaire, au vidéocodage de la facture, toutes les factures non validées.

- Dupliquez la requête par défaut existante (après l'avoir mise dans le bon classeur et avoir effectué toutes les modifications des titres des colonnes).
- Publiez cette requête dans le centre de correction (case à cocher dans la description de la requête)
- Ajoutez une condition à la requête, sur le champs FFOUR_VALIDEE=0

2. Nous allons maintenant créer la requête des factures validées nécessaire au blocage du cold, il s'agit d'une duplication de la requête des factures validées, avec quelques modifications.

- Dupliquez la requête des factures validées.
- Enlevez la case à cocher "Publier cette requête dans le centre de correction"
- Modifiez la condition à la requête, sur le champs FFOUR_VALIDEE=1

Cas d'export depuis le Cold

Préparez votre fichier template

Vous pouvez télécharger ici le zip contenant le fichier xml avec le bon nommage [exemple.zip](#)

Vous pouvez nommer le fichier avec une combinaison de valeur fixe et de valeur issue de vos champs.

La syntaxe est la suivante **[nom de champs(format)]**.

Les formats disponibles sont :

pour les nombres à virgules (.) ou (,)

pour les dates (d-m-y) ou (d.m.y) ou (y-m-d h.i.s) etc.

Notez bien que vous ne pouvez pas sortir dans des noms de fichiers, des caractères interdits (: / etc.)

Cette syntaxe est valide aussi bien dans les noms de fichiers que dans le contenu du fichier modèle.

Pour exporter des champs vides (champs non obligatoires) sans générer d'erreur vous pouvez utiliser cette syntaxe **[*FFOUR_MONTANTTTC(,)]**, l'étoile en début de champs autorise alors l'export de données vides.

Exemple de nom de fichier XML : Facture_**[FFOUR_PIECE]-[FFOUR_MONTANTTTC(,)]-[FFOUR_DATEPIECE(d/m/y)]**.xml

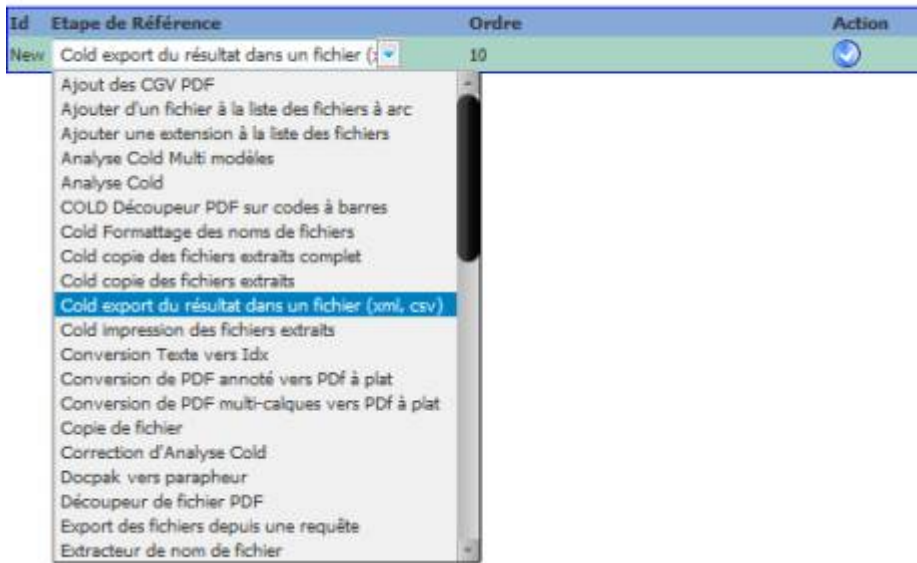
Exemple de contenu de fichier XML :

```
?<?xml version="1.0" encoding="utf-8"?>
<data>
<index id="N° Pièce">[FFOUR_PIECE]</index>
<index id="Date de la Pièce">[FFOUR_DATEPIECE(d/m/y)]</index>
<index id="Fournisseur">[FFOUR_FOURNISSEUR]</index>
<index id="Montant_TTC">[FFOUR_MONTANTTTC(.)]</index>
<index id="Echéance de Règlement"></index>
<index id="TVA Biens et Services">[FFOUR_MONTANTTVA(.)]</index>
<index id="Désignation Charge N°1"></index>
<index id="HT N°1">[FFOUR_MONTANTHT(.)]</index>
<index id="Désignation Charge N°2"></index>
<index id="HT N°2"></index>
<index id="Désignation Charge N°3"></index>
<index id="HT N°3"></index>
<index id="Port"></index>
<index id="TVA sur Immobilisations"></index>
</data>
```

Ajoutez l'étape au scenario Cold

Ajoutez cette étape :

Ajout d'une nouvelle étape au scénario



Paramétrez là de cette manière :

Étape de travail de référence sélectionnée

Id	Nom	Description	Partagé	Type
133	Cold export du résultat dans un fichier (xml, csv)	Cold Export des lignes sur la base d'un fichier modèle dans un répertoire (xml, csv), et renommage selon une charte.	<input checked="" type="checkbox"/>	SOUS PROCESSUS

Paramètres Affiliés

Id	Clé	Valeur par défaut	Valeur	Ordre
701	instance	\$instance\$	<input type="text" value="\$instance\$"/>	1
702	jobqueuid	\$jobqueuid\$	<input type="text" value="\$jobqueuid\$"/>	2
703	secuid	\$secuid\$	<input type="text" value="\$secuid\$"/>	3
704	coldmdlid	0	<input type="text" value="22 - facturesfournisseurs"/>	4
705	outputpath		<input type="text" value="f:\temp\[FFOUR_FOURNISSEUR]"/>	5
706	templatefile		<input type="text" value="F:\[FFOUR_PIECE]-[FFOUR_MONTANTTTC(,)]-[FFOUR_DATEPIECE(d-m-Y)].xml"/>	6
707	qryid		<input type="text" value="36 - facturesfournisseurs"/>	7
708	encoding	iso8859-15	<input type="text" value="utf-8"/>	8

coldmdlid = N° de modèle cold (normalement sélectionné automatiquement)

outputpath = chemin de sortie , le chemin peut comporter des champs et des backslashes, les répertoires se créeront automatiquement

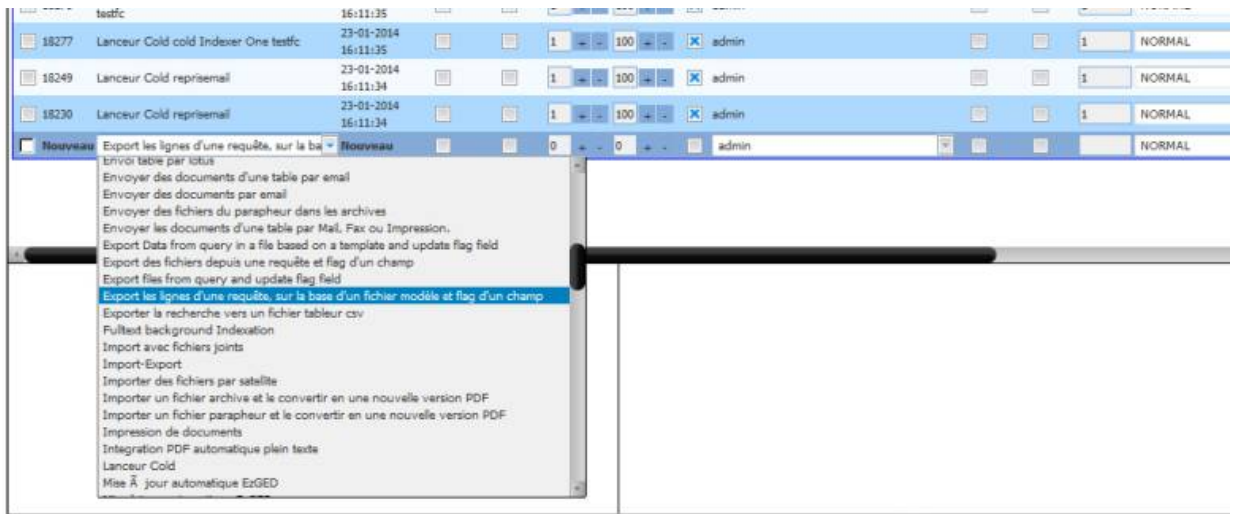
template = chemin complet du fichier xml exemple contenant les balises des champs, aussi bien dans le nom du fichier que à l'intérieur du fichier.

qryid = Numéro de la requête factures validées, qui conditionne l'export des datas, uniquement si les factures sont validées dans le centre d'indexation. En attendant cette condition, le fichier reste bloqué en erreur à l'étape.

encoding = encodage du fichier de sortie ici utf-8 pour le xml, le plus souvent iso8859-15. Attention si il s'agit de utf-8, éditez bien votre fichier template en utf-8, avec Notepad++ par exemple.

Cas d'export depuis une requête

- Préparez votre fichier template (même étape que export depuis le cold)
- Créez le travail qui scrute la table



- **Créez le travail de type** : export des lignes d'une requête, sur la base d'un fichier modèle et flag d'un champs

priorité = 1

poinds = 100

etape = 1

actif = laissez décoché pour l'instant

purgeable = jamais

type= systeme

- **Paramétrez les champs suivants**

Il faut créer chaque champ dans le formulaires, accessible sur le nouveau travail par l'icône paramètres

Paramètres pour le travail sélectionné

Id	Clé	Valeur	Ordre
158473	flagfield	FFOUR_EXPORTEE	1
158474	outpath	f:\temp\[FFOUR_FOURNISSEUR]	2
158475	templatefile	f:\[FFOUR_PIECE]_[FFOUR_MONTANTTTC(,)]_[FFOUR_DATEPIECE(d-m-Y)]_0.xml	3
158476	qryid	36	4
158477	encoding	utf-8	5
Nouveau			6

le **qryid** correspond à l'id de la requête factures validées, et le **flagfield** correspond au champs qui sert de contrôle des lignes à exporter, il est mis à vrai une fois la ligne exportée correctement.

- **Ensuite programmez le travail**, à l'aide de l'icône programmer, et modifier la case actif du travail pour qu'il s'exécute.

From: <http://wiki.ezdev.fr/> - EzGED Wiki

Permanent link: http://wiki.ezdev.fr/doku.php?id=doc:admin:cold:export_xml&rev=1390818284

Last update: 2023/03/17 09:56

