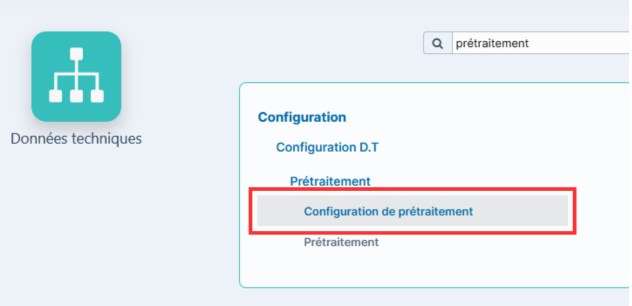
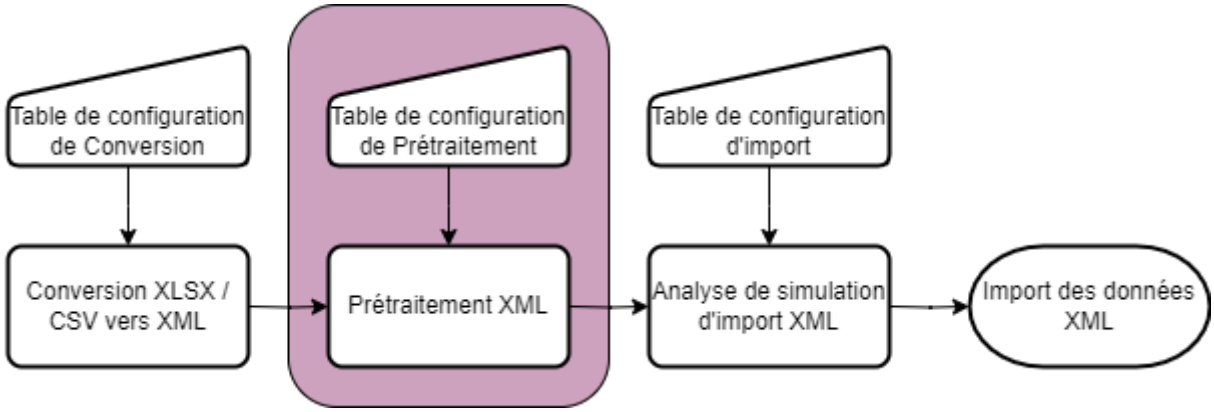
Paramétrage du prétraitement des données XML

Nous allons maintenant nous concentrer sur la configuration du prétraitement des données XML. Pour cela rendez vous dans le menu **Données techniques > Configuration D.T > Configuration prétraitement**.



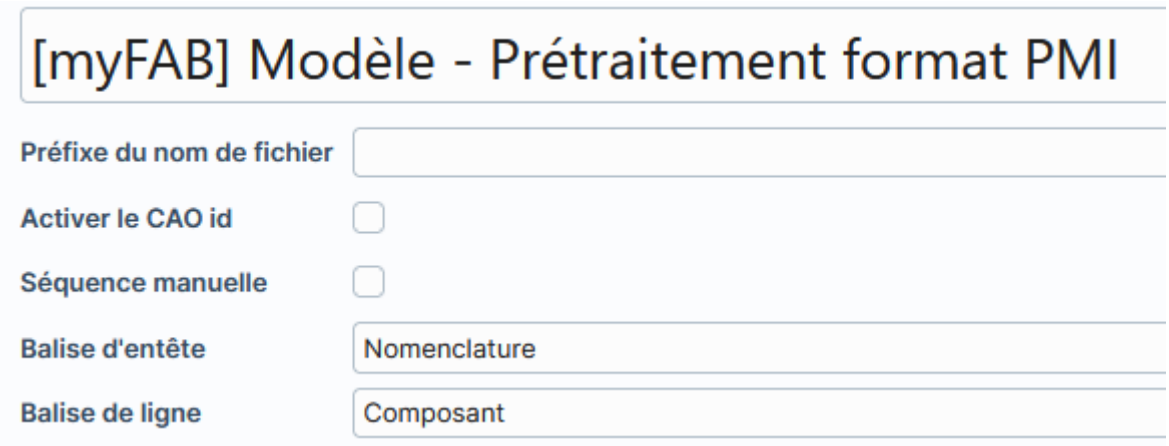
Le prétraitement du fichier d'import permet de réaliser plusieurs actions visant à modifier le format du fichier et plus particulièrement de ses balises afin de le rendre prêt à être importé dans Open-Prod.

Un traitement d'import appelle un prétraitement. Ce prétraitement appelle lui-même une règle de prétraitement, mais aussi une règle de conversion. Cela permet de regrouper les différents flux au même niveau et d'avoir un traitement global appelant différents traitements enfants.



1. Balises de prétraitement

Une configuration de prétraitement est appelée au travers d'un prétraitement.



Préfixe du nom de fichier : lors du prétraitement, le fichier est renommé automatiquement. Lorsque ce paramètre est laissé vide, le fichier est renommé par défaut avec le préfixe "Pretreatement-". Il est possible de forcer un préfixe via ce champ.

Activer le CAO id : Permet d'incrémenter une balise "CaoID" dans les balises dans certains cas. Il s'agit de la clé de la balise identifiant de manière unique les composants (ID, code produit...). Comme un même produit peut être appelé à plusieurs reprises et à plusieurs niveaux dans un même fichier d'import, il est nécessaire d'avoir une clef rendant unique chaque élément. Le code produit n'est pas suffisant. Grâce à cette clef de balise de ligne, le système va pouvoir attribuer automatiquement un identifiant unique appelé "CaoID". Cet identifiant peut ensuite être utilisé dans les domaines de rechercher du traitement d'import. Vous pouvez laisser vide cette case

Balise d'entête : il s'agit de la balise permettant d'identifier la nomenclature.

Balise de ligne : il s'agit de la balise permettant d'identifier les composants des nomenclatures.

Séquence manuelle : si cette option est cochée, cela signifie que le fichier à importer contient déjà un séquençage de la composition des données techniques. Si l'option n'est pas cochée, le système va s'appuyer sur les paramètres ci-dessus pour attribuer une séquence aux différentes lignes traitées (dans les lignes de nomenclatures et de gammes notamment).

2. Règles de prétraitement

Il est possible d'ajouter ou de supprimer des règles de prétraitement qui seront exécutées dans l'ordre de leur séquence.

Règles

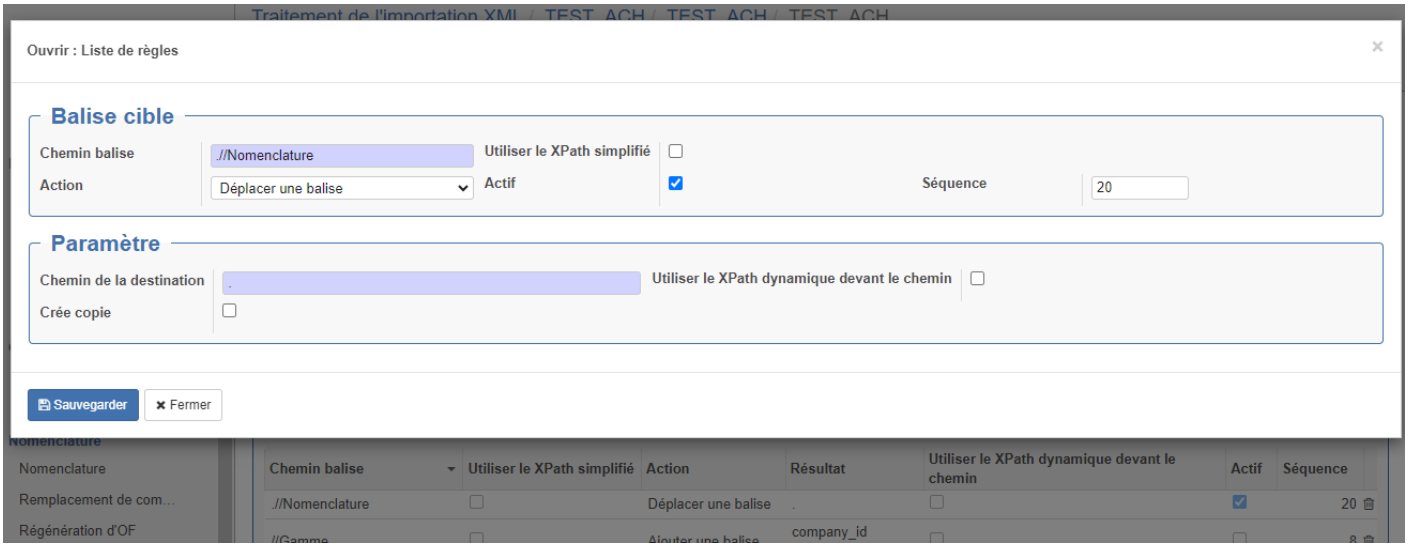
1-13 sur 13

Chemin balise	Utiliser le XPath simplifié	Action	Résultat	Utiliser le XPath dynamique devant le chemin	Actif	Séquence
.//Nomenclature	☐	Déplacer une balise	.	☐	☑	20
//Gamme	☐	Ajouter une balise	company_id avec []	☐	☐	8
//Ligne	☐	Ajouter une balise	company_id avec []	☐	☐	6
//Ligne/Nomenclature/Produit	☐	Déplacer une balise	/../.. Avec copie	☑	☑	19
//Ligne_gamme	☐	Ajouter une balise	company_id avec []	☐	☐	7
//Nomenclature	☐	Ajouter une balise	company_id avec []	☐	☐	5
//Nomenclature/Produit/CaoID	☐	Déplacer une balise	/../..	☑	☑	80
//categ_name	☐	Déplacer une balise	//categ_ressource	☑	☐	61
//categ_ressource_lg	☐	Ajouter une balise	categ_ressource avec ['categ_name']	☐	☑	60
//company_id	☐	Ajouter une balise	//company_id_name avec []	☐	☐	62
//company_id_name	☐	Ajouter un attribut	value = SOC	☑	☐	63
//quantity	☐	Déplacer une balise	/../..	☑	☑	50
Ligne/Produit	☑	Renommer une balise	Composant	☐	☑	70

Ajouter un élément

En cliquant sur une des lignes, vous pouvez accéder à la modification de la règle. En cliquant sur "Ajouter un élément" en bas de la fenêtre, vous pouvez créer une nouvelle règle.

Le formulaire de saisie "Ouvrir : Liste de règles" s'ouvre alors.



Ouvrir : Liste de règles

Balise cible

Chemin balise	//Nomenclature/Produit/CaoID	Utiliser le XPath simplifié	☐		
Action	Déplacer une balise	Actif	☑	Séquence	10

Paramètre

Chemin de la destination	/../..	Utiliser le XPath dynamique devant le chemin	☑
Crée copie	☐		

× Fermer

Ouvrir : Liste de règles

Balise cible

Chemin balise	//Nomenclature/Produit/Code	Utiliser le XPath simplifié	☐		
Action	Déplacer une balise	Actif	☑	Séquence	25

Paramètre

Chemin de la destination	/../..	Utiliser le XPath dynamique devant le chemin	☑
Crée copie	☑		

× Fermer

Ouvrir : Liste de règles

Balise cible

Chemin balise	//Nomenclature/Indice	Utiliser le XPath simplifié	☐		
Action	Renommer une balise	Actif	☑	Séquence	21

Paramètre

Nom de la nouvelle balise	Version_Nomenc
Utiliser le XPath dynamique devant le chemin	☐

× Fermer

Ouvrir : Liste de règles

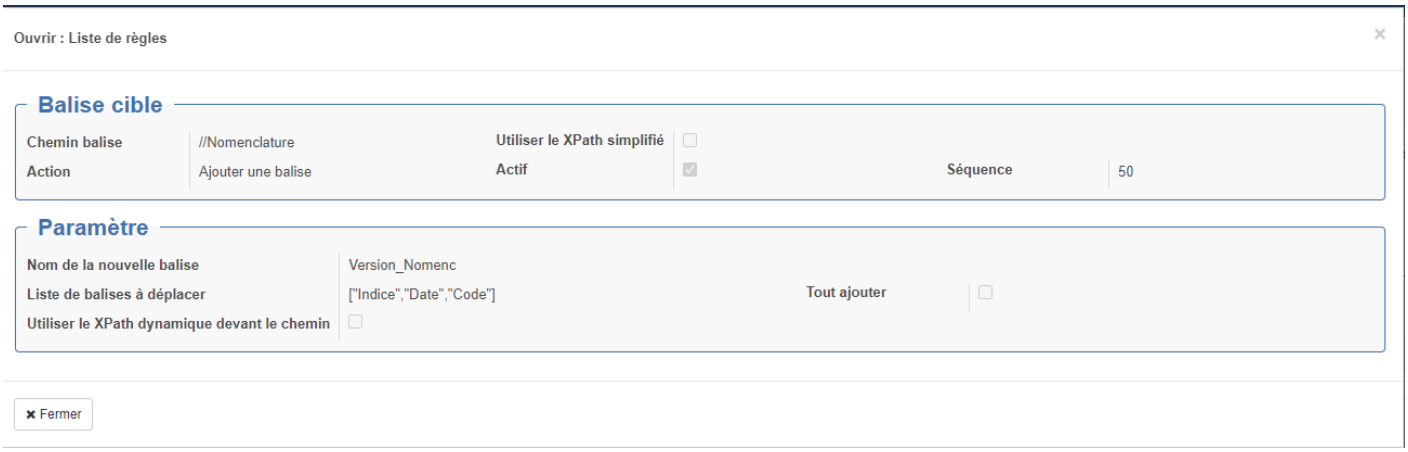
Balise cible

Chemin balise	//Nomenclature/Version	Utiliser le XPath simplifié	☐		
Action	Supprimer une balise	Actif	☑	Séquence	40

Paramètre

Effacer le contenu	☐
Utiliser le XPath dynamique devant le chemin	☐

× Fermer



Section "Balise cible" :

- Chemin balise : Emplacement de la balise à modifier (voir ci-après des explications sur les syntaxes utilisables)
- Action : Action réalisée
  - Ajouter une balise
  - Fusionner des balises
  - Supprimer une balise
  - Renommer une balise
  - Ajouter un attribut
  - Déplacer une balise
  - Modifier un attribut
  - Concaténation d'attributs
  - Supprimer balise avec condition
- Utiliser le XPath simplifié : exemple de simplifié : Ligne/Produit exemple de xpath non simplifié : //Ligne/Nomenclature/Produit ou .//Nomenclature
- Actif : si cette case n'est pas coché, l'action ne s'exécutera pas
- Séquence : permet de gérer l'ordre dans le quel les actions seront exécutées.

Section "Paramètre" :

- Chemin de la destination : emplacement relatif de destination (dans le cas d'un déplacement par exemple) : exemple "/.." permet de remonter d'un niveau

- Créer copie : en cas d'un déplacement, laisse la balise d'origine à sa place
- Nom de la nouvelle balise : Nom de la nouvelle balise
- Liste de balises à déplacer : Liste des balises à créer dans le cas d'un ajout de balise exemple : ["Indice","Date","Code"]
- Effacer le contenu : en cas de suppression de balise, efface les balises dans celle-ci
- Utiliser le XPath dynamique devant le chemin : permet d'avoir un chemin de destination sous forme d'un Xpath dynamique

Si l'action concerne une manipulation des attributs, les paramètres suivants sont affichés :

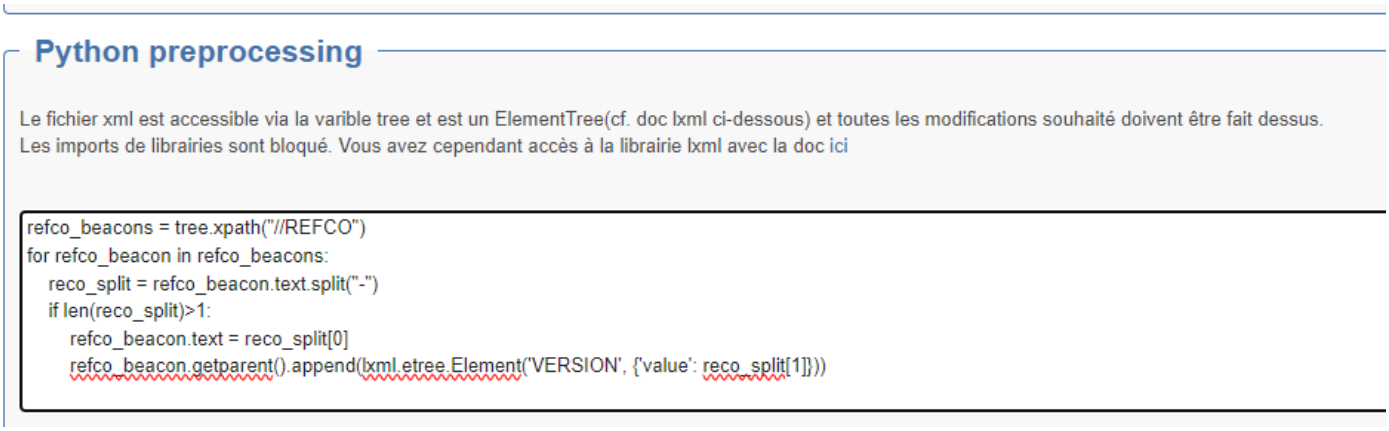
- Nom de l'attribut : Nom de l'attribut
- Valeur de l'attribut (nouvelle et / ou ancienne) : Valeur de l'attribut
- Attribut à concaténer :Attribut à concaténer
- Balise à concaténer :Balise à concaténer
- Séparateur : Séparateur à mettre entre les valeurs

3. Prétraitement Python (Python preprocessing)

La zone de Traitement python permet de saisir du code python venant s'exécuter à la fin du prétraitement, cela va permettre gérer beaucoup de cas trop compliqué à gérer via l'interface.

Le fichier xml est accessible via la varible tree et est un ElementTree(cf. doc lxml ci-dessous) et toutes les modifications souhaité doivent être fait dessus.

Les imports de librairies sont bloqué. Vous avez cependant accès à la librairie lxml avec la doc ici



Exemples de pré-traitement :

Exemple 1 : dans la balise REFCO supprimer ce qui est après le tiret et créer une nouvelle balise VERSION avec ce qui est apès le - Avant : <REF>1144-02</REF> -> vers <REF>1144</REF>

```
refco_beacons = tree.xpath("//REFCO")
for refco_beacon in refco_beacons:
    reco_split = refco_beacon.text.split("-")
    if len(reco_split)>1:
        refco_beacon.text = reco_split[0]
```

```
        refco_beacon.getparent().append(lxml.etree.Element('VERSION', {'value':
reco_split[1]}))
```

Exemple 2 : Ajouter balise CATEG avec attribut value="KIT" si contenu balise LIB commence par "INSTALLATION" sinon value="STANDARD"

```
lib_beacons = tree.xpath("//LIB")
for lib_beacon in lib_beacons:
    if "INSTALLATION" in lib_beacon.text:
        lib_beacon.getparent().append(lxml.etree.Element('CATEG', {'value': "KIT"}))
    else:
        lib_beacon.getparent().append(lxml.etree.Element('CATEG', {'value': "STANDARD"}))
```

Exemple 3 : mettre le contenu des balises REF et REFCO en attribut "value" (<REF>111</REF> -> vers <REF value="111" />

```
ref_beacons = tree.xpath("//REF")
refco_beacons = tree.xpath("//REFCO")
for ref_beacon in ref_beacons:
    ref_beacon.set("value", ref_beacon.text)
    ref_beacon.text=""
for refco_beacon in refco_beacons:
    refco_beacon.set("value", refco_beacon.text)
    refco_beacon.text=""
```

Exemple 4 : supprimer balises si vide et si different de catgorie

```
to_delete_beacons = tree.xpath("/ARTICLES/ARTICLE/*")
balise_a_garder =["Categorie","REF","Libelle"]
for to_delete_beacon in to_delete_beacons :
    if (to_delete_beacon.text in ["","  "] or not to_delete_beacon.text) and
to_delete_beacon.tag not in balise_a_garder :
        to_delete_beacon.getparent().remove(to_delete_beacon)
```

Exemple 5 : si le LIB_ASS commence par installation ou mise en route et que code produit commence par D alors Catégorie KIT // récupère la catégorie de l'article s'il existe déjà.

```
LIB_ASS_beacons = tree.xpath("//LIB_ASS")
for LIB_ASS_beacon in LIB_ASS_beacons :
    Categ_name_beacon=LIB_ASS_beacon.xpath("../Categorie/Categ_name")[0]
    code_produit=LIB_ASS_beacon.xpath("../REF")[0].text
```

```
    product_rc=env['product.product'].search([('code','=',code_produit)], limit=1)
    if product_rc :
        Categ_name_beacon.text=product_rc.categ_id.name
    elif LIB_ASS_beacon.xpath("../REF")[0].text.startswith("D") and
(LIB_ASS_beacon.text.startswith("INSTALLATION" ) or LIB_ASS_beacon.text.startswith("MISE EN
ROUTE")  ) :
        Categ_name_beacon.text="KIT"
    else :
        Categ_name_beacon.text="NEW PRODUCTS"
```

Lxml - 2 façons de parcourir l'object :

· *Via une boucle for comme une liste :*
for child in tree :
child.get('value')
o Pour parcourir tous les enfants directs de tree

· *Via un xpath :*
child = tree.xpath("/Sequence")
child.get('value')
o Pour chercher toutes les balises Sequence qui sont enfant direct de tree

Ajouter une balise :
tree.append(lxml.etree.Element('Sequence', {'value': 1}))

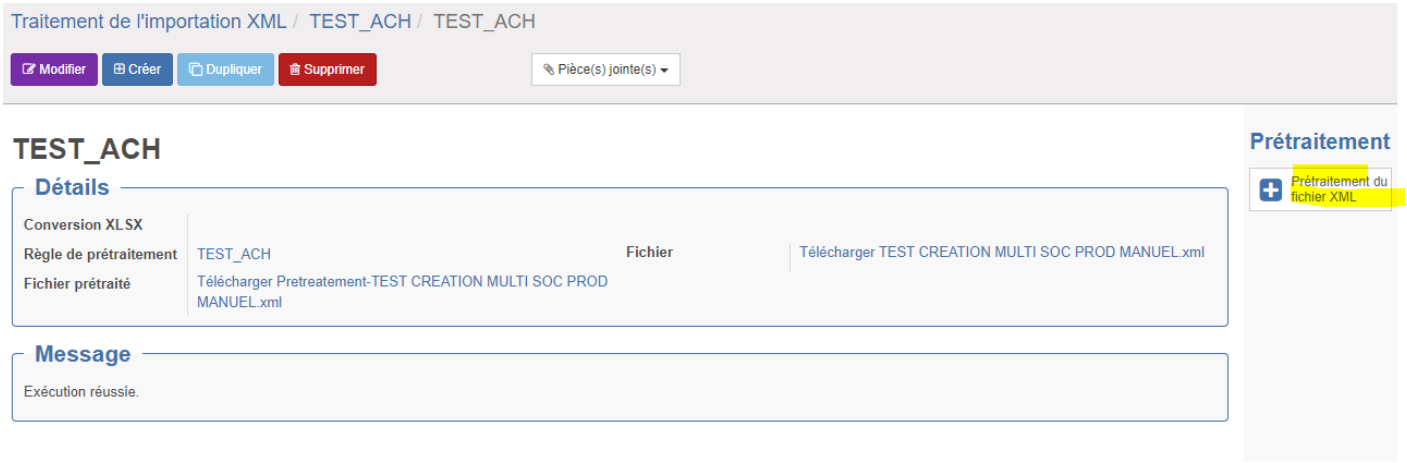
Pour ajouter une balise Sequence avec en value 1 dans les enfants de tree :

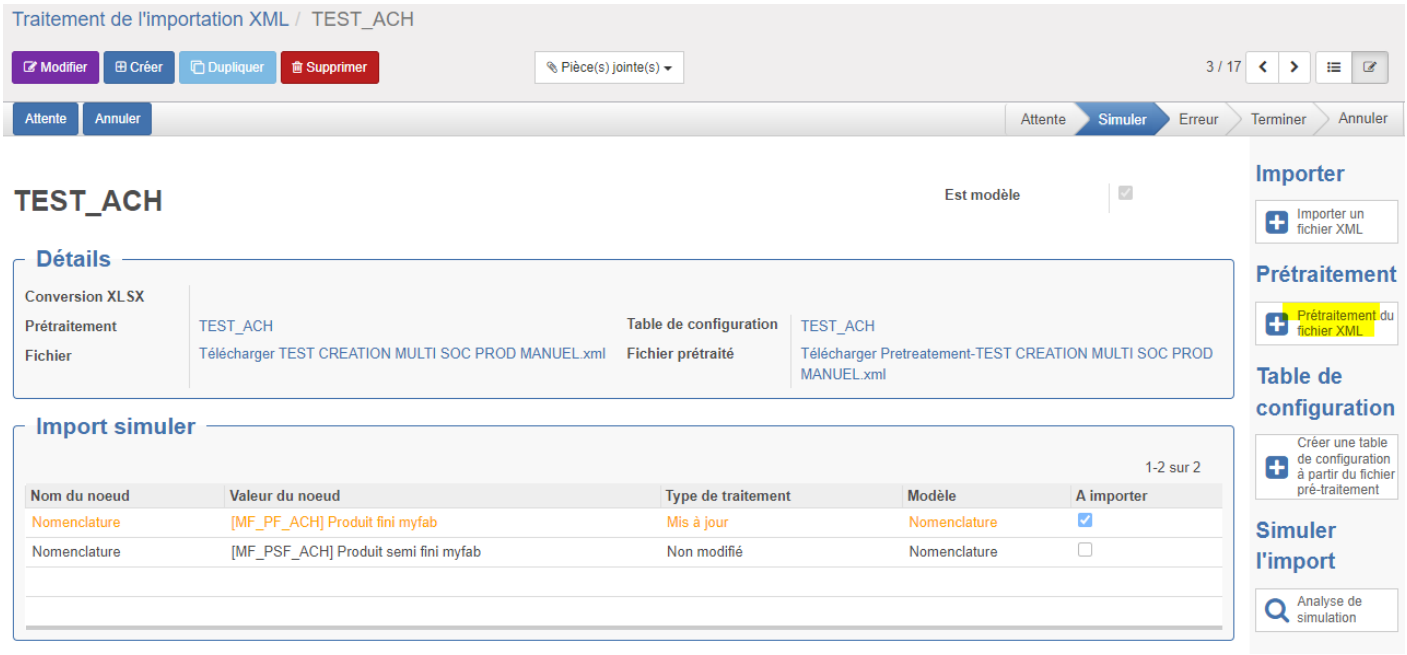
```
<tree>
    <Sequence value="1" />
</tree>
```

Retirer une balise :
Sequence_children = tree.xpath("/Sequence")
for child in Sequence_children :
child.getparent().remove(child)

4. Exécution du prétraitement

Le prétraitement peut s'effectuer depuis le traitement d'import ou bien de la fenêtre de "Prétraitement xml". Dans les deux cas, il faut cliquer sur le bouton d'action "Prétraitement du fichier XML". Il s'effectuera sur base de la règle de prétraitement sélectionnée.

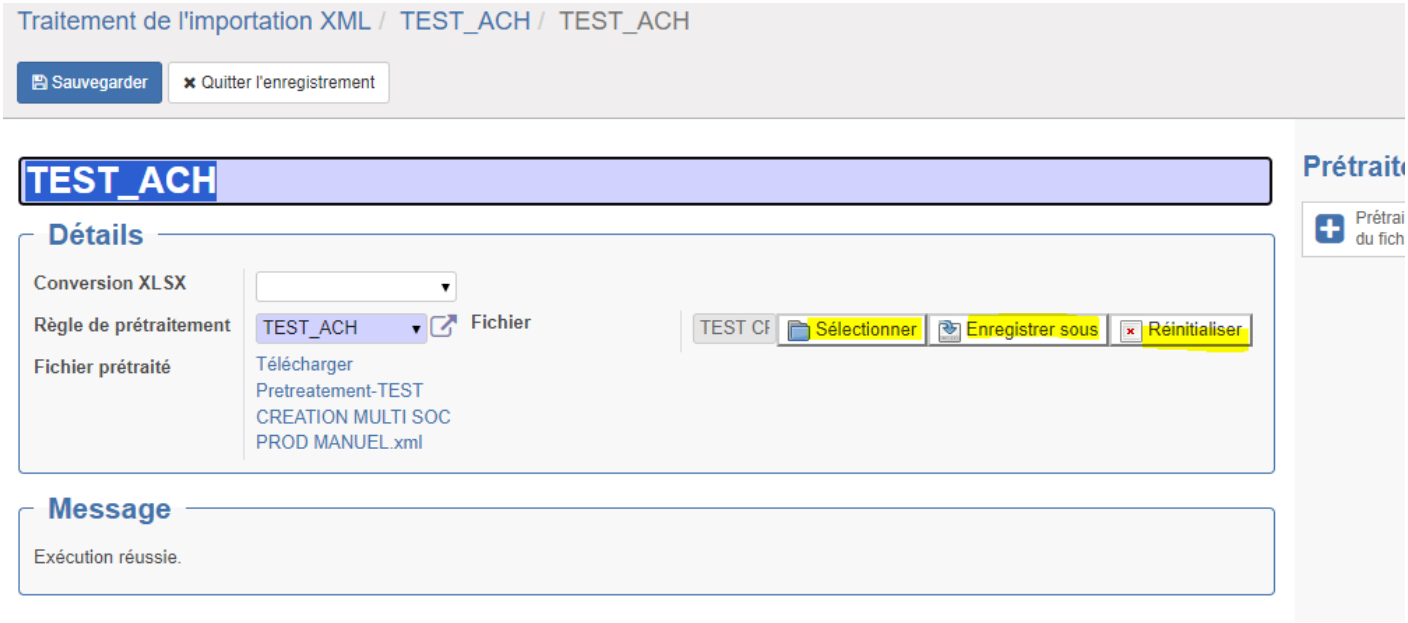




Bien entendu, il est nécessaire d'avoir un fichier au **format XML** à prétraiter. Pour cela, il suffit d'entrer en modification puis de cliquer sur "Sélectionner" pour choisir un fichier à prétraiter.

Si une règle de conversion XLSX est sélectionnée, il faudra sélectionner un fichier .xlsx puis le convertir en fichier .xml pour ensuite le prétraiter.

Le bouton "Enregistrer sous" sert à télécharger le fichier .xml sélectionné ou converti. Enfin, le bouton "Réinitialiser" permet de vider le champ de tout fichier.



La section "Message" affiche le résultat du dernier prétraitement réalisé. Si une erreur est survenue, elle est écrite ici. A chaque nouvelle exécution du prétraitement, le message est remplacé. Aussi, si deux prétraitement réussissent à la suite, le message "Exécution réussie" semble ne pas changer.

5. Glossaire sur les syntaxes utilisables dans les xpath.

1 - La syntaxe

La syntaxe de XPath correspond aux expressions utilisées pour sélectionner des nœuds. Ces expressions sont essentielles pour spécifier les chemins empruntés dans les documents XML.

Les expressions couramment utilisées sont listées ci-dessous :

nodename	Sélectionner les nœuds avec le nom « nodename » (à remplacer par le nom du nœud recherché)
/	Sélectionner le nœud racine
//	Sélectionner les nœuds dans le document, peu importe leur position
.	Sélectionner le nœud actif
..	Sélectionner le nœud parent du nœud actif
@	Sélectionner les nœuds de type attribut
::	Spécifier un axe
*	Sélectionner les nœuds contextuels de type élément

L'utilisation du caractère / en début d'expression spécifie le contexte actuel comme étant la racine du document, c'est une expression absolue. À l'inverse, une expression ne commençant pas par le caractère / est une expression relative à un contexte.

Pour mieux comprendre la syntaxe des expressions XPath, il est préférable de s'appuyer sur des exemples concrets. Ainsi, l'exemple suivant de fichier XML va faire office de fil conducteur pour illustrer l'ensemble des principes de XPath. Ce fichier XML rassemble des informations sur des paires de chaussettes et des bonnets (les tailles, les prix et les matières) rangés dans plusieurs tiroirs nommés t1 et t2.

Un exemple de document XML pour illustrer l'extraction de données avec XPath.

Par exemple, l'expression /armoire renvoie à toutes les valeurs présentes dans l'armoire à la racine. Ici, les données retournées sont Chaussettes Sport, Chaussettes Ville, Chaussettes décontractées, Chaussettes Hiver, Bonnet Mi-saison, Bonnet Pluie.

L'expression /armoire/tiroir/chaussettes sélectionne toutes les valeurs chaussettes présentes dans chaque tiroir.

L'expression //chaussettes/@taille sélectionne l'attribut taille de toutes les valeurs chaussettes présentes n'importe où dans le document XML. Ici, les valeurs retournées sont 40, 42, 38, 44.

2 - Les nœuds

Les documents XML sont traités comme des arbres de nœuds. Il existe sept types de nœuds : les éléments, les attributs, les nœuds de texte, les espaces de noms, les instructions de traitement, les commentaires et le nœud racine.

Pour sélectionner le type de nœud souhaité, il est nécessaire d'utiliser un filtre, aussi appelé test de nœud. Il est important de noter que les attributs et les espaces de nom sont sélectionnés à partir d'axes.

Les tests de nœud les plus utilisés sont :

node()	Correspond à n'importe quel nœud
text()	Correspond à n'importe quel nœud de texte
comment()	Correspond à n'importe quel nœud de commentaire
*	Correspond à n'importe quel nœud élément
processing-instruction('cible')	Correspond aux nœuds d'instructions de traitement

En se basant sur l'exemple du fichier XML précédent, le test de nœud //text() sélectionne toutes les valeurs textuelles présentes dans le document, peu importe la position. Ici les valeurs retournées sont Chaussettes Sport, Chaussettes Ville, Chaussettes décontractées, Chaussettes Hiver, Bonnet Mi-saison, Bonnet Pluie.

Le test de nœud //comment() sélectionne tous les commentaires du document, ici Commentaire : Armoire avec deux tiroirs de chaussettes, différentes matières.

3 - Les axes

Les axes représentent les relations généalogiques entre les nœuds. Ils sont utilisés pour localiser et sélectionner des nœuds relatifs aux nœuds contextuels, que ce soit des nœuds parents, enfants ou frères.

Les axes couramment utilisés sont :

self	Sélectionner le nœud actif
child	Sélectionner les nœuds enfants du nœud actif
parent	Sélectionner le nœud parent du nœud actif
ancestor, ancestor-or-self	Sélectionner les nœuds ancêtres du nœud actif
descendant, descendant-or-self	Sélectionner les nœuds descendants du nœud actif
preceding	Sélectionner les nœuds qui précèdent le nœud actif
following	Sélectionner les nœuds qui suivent le nœud actif
preceding-sibling	Sélectionner les nœuds frères qui précèdent le nœud actif
following-sibling	Sélectionner les nœuds frères qui suivent le nœud actif
namespace	Sélectionner les nœuds de type espace de nom
attribute	Sélectionner les nœuds de type attribut

Voici quelques exemples concrets basés sur le fichier XML précédent :

L'expression /armoire/descendant::chaussettes sélectionne tous les descendants chaussettes de l'élément racine armoire, ici Chaussettes Sport, Chaussettes Ville, Chaussettes Décontractées, Chaussettes Hiver.

L'expression //chaussettes/following-sibling::bonnet sélectionne les nœuds frères bonnet qui suivent les nœuds chaussettes, ici Bonnet Mi-saison, Bonnet Pluie.

4 - Les prédicats

L'utilisation des axes ne suffit pas à spécifier la position d'un nœud, afin d'obtenir un résultat plus précis, il est conseillé d'utiliser les prédicats. Les prédicats permettent de filtrer les nœuds sélectionnés par les axes et les tests de nœud. Les prédicats sont intégrés dans l'expression à l'aide de crochets []. Les valeurs indiquées dans les prédicats correspondent aux nœuds de type attribut.

Par exemple, l'expression //tiroir[@id='t1']/chaussettes[1]/following-sibling::chaussettes sélectionne les nœuds frères qui suivent la première paire de chaussettes présente dans le tiroir ayant pour identifiant t1. Ici, le résultat est Chaussettes Ville.

L'expression //chaussettes[@taille="42"]/@matiere renvoie vers l'attribut matiere des paires de chaussettes ayant pour taille 42. Ici, le résultat est Viscose.

> *Sources https://blog.hubspot.fr/website/xpath*

Autre documentation en anglais en anglais

Revision #21
Created 21 November 2022 14:23:01 by Alexis CHAPEL
Updated 20 April 2026 13:18:52 by Marius DURAND