Scribe - Tâche #37590

Scénario # 36204 (Terminé (Sprint)): Ajout de fonctionnalités dans Edition Groupée EAD : changement de classe

etude

17/08/2026 15:22 - Ludwig Seys

Statut:	Fermé	**Début:**	01/10/2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Ludwig Seys	**% réalisé:**	100%
Version cible:	Livraison Cadoles - MEN 31/07/2026 (45)	**Temps estimé:**	0.00 heure

Description

Historique

#1 - 17/08/2026 15:22 - Ludwig Seys

- Statut changé de Nouveau à En cours

#2 - 17/08/2026 15:22 - Ludwig Seys

Ajouter dans :

EAD → Gestion → Édition groupée → Utilisateurs

Le fonctionnement attendu doit s'inspirer de l'action existante : Inscrire ces utilisateurs à d'autres groupes avec les adaptations nécessaires pour :

- ne traiter que les élèves ;
- utiliser les fonctions backend existantes de changement de classe ;
- prendre en compte le mode multi-établissement ;
- éviter les synchronisations inutiles lors d'un traitement en masse ;
- ne pas modifier `scribe-backend` si les primitives existantes sont suffisantes.

Dépôt : ead
Paquet installé : eole-ead-server

Le fichier installé correspondant est : /usr/share/ead2/backend/actions/scribe/grouped_edition.py

La modification fonctionnelle devrait être réalisée principalement dans le dépôt **ead**.

Backend Scribe

Dépôt : scribe-backend
Paquet : eole-scribe-backend

Le backend contient notamment :

- scribe/eleves.py
- scribe/eoleuser.py
- scribe/eolegroup.py
- scribe/importation/writer.py

L'étude du dépôt montre que les fonctions nécessaires au changement de classe existent déjà.
À ce stade, aucune modification de `scribe-backend` ne semble nécessaire.

#3 - 18/08/2026 10:06 - Ludwig Seys

Fonctionnement actuel de l'édition groupée EAD

Le point d'entrée est : ead/backend/actions/scribe/grouped_edition.py
La liste des formulaires acceptés est déclarée dans : form_result = Dict(...)

La validation est dispatchée dans : def _valid_form(self):

les action existantes sont utilisées comme modèle

Le bouton : => Inscrire ces utilisateurs à d'autres groupes est créé dans : ead/backend/actions/scribe/grouped_edition.py
dans : def _get_action_btns(self, num=''):
Le chargement du formulaire est assuré par : def _get_action_form(self, _type):

Formulaire « inscription à un groupe »

Le formulaire correspondant se trouve dans : ead/backend/actions/scribe/tool/grpedit.py
fonction : def get_inscription_form(server_nb, name, role):
Le nouveau changement de classe peut reprendre ce fonctionnement avec un nouveau formulaire.

Template EAD concerné

L'affichage des formulaires se trouve dans : ead/backend/template/scribe_edition.tmpl
Le formulaire d'inscription à un groupe est actuellement affiché à partir de : subscribe_to_group

Gestion des index des boutons

Les boutons de l'édition groupée sont actuellement indexés :

0 : inscription groupe
1 : quota
2 : domaine
3 : profil
4 : mot de passe
5 : shell
6 : rôle (administrateur uniquement)

Ces index sont utilisés par : ead/backend/actions/scribe/tool/grpedit.py
dans : get_toexec_links()

L'ajout du nouveau bouton devra donc prendre en compte ces index afin d'éviter des effets de bords.

Changement de classe individuel existant

Le changement de classe existe déjà dans l'édition individuelle d'un élève.

Fichier : ead/backend/actions/scribe/usermodify.py
Il n'est donc pas nécessaire de réimplémenter dans EAD les opérations LDAP/Samba associées à un changement de classe, surtout que l'existant gère déjà le mutli-établissement.

Backend de changement de classe

La fonction utilisée se trouve dans : scribe-backend/scribe/eleves.py
plus précisément : def _change_classe(self, user, new_classe, new_etab=None, sync=True):

Elle réalise déjà les opérations nécessaires.

Les attributs modifiés comprennent notamment :
- Divcod
- Meflcf
- ENTEleveClasses
- ENTEleveMEF
- ENTEleveLibelleMEF

Les désinscriptions sont effectuées via : self._desinscription(...)
et les inscriptions via : self._inscription(...)

La fonction est donc adaptée à la fonctionnalité demandée.

Comportement pour les utilisateurs non élèves

L'action doit concerner les élèves uniquement.
Une sélection EAD pouvant contenir plusieurs types d'utilisateurs, le comportement retenu est pour exemple :

20 utilisateurs sélectionnés |
+-- 17 élèves -> changement de classe |
+-- 3 non-élèves -> ignorés

Les utilisateurs non élèves ne doivent donc pas faire échouer l'ensemble du traitement.
L'identification peut être réalisée avec la classe déjà disponible : from scribe.eleves import Eleve.
Un message de résultat devra idéalement indiquer le nombre d'élèves modifiés et le nombre d'utilisateurs ignorés.

Si aucun élève n'est présent dans la sélection, l'opération devra retourner un message explicite plutôt qu'un faux succès.

Étude du mode multi-établissement

Le backend supporte explicitement le multi-établissement.
En multi-établissement, il faut en revanche éviter qu'un élève appartenant à un établissement secondaire soit traité implicitement avec l'établissement principal.

Protection existante contre les changements inter-établissements

Le backend contient déjà une vérification dans : scribe/eoleuser.py
fonction : _inscription(...)

En mode multi-établissement :

```
if SUPPORT_ETAB:
    etabgroup = self.get_etab_from_group(groupe)

    if check_etab:
        if etab is None:
            etabuser = self.get_etab(login)
        else:
            etabuser = etab

        if etabgroup != etabuser:
            raise Exception(...)
```

Une inscription dans un groupe ou une classe appartenant à un autre établissement est donc déjà détectée par le backend.

Traitement en masse déjà utilisé par Scribe

Le backend d'importation des élèves fournit déjà un exemple de traitement massif.
Fichier : scribe-backend/scribe/importation/writer.py
Lors d'un changement de classe :

```
user._change_classe(
    login,
    classe,
    etab,
    sync=False
)
```

Puis, lorsque tous les élèves ont été traités :

```
for login in userlist:
    user._gen_ftpdir(login)
    user._gen_groupesdir(login)
```

Le backend lui-même montre donc que le mode : sync=False est prévu pour les traitements en masse.
Cette approche semble adaptée à l'édition groupée EAD.

Fichiers EAD à modifier

Les modifications devraient rester limitées principalement à trois fichiers.

backend/actions/scribe/grouped_edition.py
backend/actions/scribe/tool/grpedit.py
backend/template/scribe_edition.tmpl

Fichiers étudiés mais ne nécessitant a priori pas de modification

Dépôt `scribe-backend` :

- scribe/eleves.py
- scribe/eoleuser.py
- scribe/eolegroup.py
- scribe/importation/writer.py
- scribe/ldapconf.py

Ces fichiers fournissent :

- `_change_classe()` ;
- `_inscription()` ;
- `_desinscription()` ;
- `get_etab()` ;
- `get_etab_from_group()` ;
- `get_classes(etab=...)` ;
- les protections inter-établissements ;
- le fonctionnement `sync=False` pour les traitements massifs.

Aucune modification de ce dépôt n'est actuellement identifiée comme nécessaire.

#4 - 18/08/2026 10:06 - Ludwig Seys

- Statut changé de En cours à Résolu

#5 - 07/09/2026 15:58 - Joël Cuissinat

- Statut changé de Résolu à Fermé

- % réalisé changé de 0 à 100

- Restant à faire (heures) mis à 0.0

Très belle étude ;)