

## Authentic 2 - Bug #19168

### Erreur d'encodage lors synchro ldap.

03 octobre 2017 12:07 - Josué Kouka

<b>Statut:</b>	Rejeté	<b>Début:</b>	03 octobre 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Paul Marillonnet (retour le 15/04)	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		

#### Description

```
Traceback (most recent call last):
  File "/usr/lib/authentic2/manage.py", line 21, in <module>
    execute_from_command_line(sys.argv[1:] + argv)
  File "/usr/lib/python2.7/dist-packages/django/core/management/__init__.py", line 354, in execute_from_command_line
    utility.execute()
  File "/usr/lib/python2.7/dist-packages/django/core/management/__init__.py", line 346, in execute
    self.fetch_command(subcommand).run_from_argv(self.argv)
  File "/usr/lib/python2.7/dist-packages/hobo/multitenant/management/commands/tenant_command.py", line 52, in run_from_argv
    klass.run_from_argv(args)
  File "/usr/lib/python2.7/dist-packages/django/core/management/base.py", line 394, in run_from_argv
    self.execute(*args, **cmd_options)
  File "/usr/lib/python2.7/dist-packages/hobo/authentic2/apps.py", line 45, in new_execute
    return old_execute(self, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/django/core/management/base.py", line 445, in execute
    output = self.handle(*args, **options)
  File "/usr/lib/python2.7/dist-packages/authentic2/management/commands/sync-ldap-users.py", line 14, in handle
    list(LDAPBackend.get_users())
  File "/usr/lib/python2.7/dist-packages/authentic2/backends/ldap_backend.py", line 847, in get_users
    users = conn.search_s(user_basedn, ldap.SCOPE_SUBTREE, user_filter, attrlist=attrs)
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 552, in search_s
    return self.search_ext_s(base, scope, filterstr, attrlist, attrsonly, None, None, timeout=self.timeout)
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 870, in search_ext_s
    return self._apply_method_s(SimpleLDAPObject.search_ext_s, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 812, in _apply_method_s
    return func(self, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 545, in search_ext_s
    msgid = self.search_ext(base, scope, filterstr, attrlist, attrsonly, serverctrls, clientctrls, timeout, sizelimit)
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 541, in search_ext
    timeout, sizelimit,
  File "/usr/lib/python2.7/dist-packages/ldap/ldapobject.py", line 99, in _ldap_call
    result = func(*args, **kwargs)
UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 25: ordinal not in range(128)
```

<https://sentry.entrouvert.org/sentry/recette/issues/2297/>

#### Demandes liées:

Lié à Authentic 2 - Bug #23698: LDAP: supporter les accents dans les base DNs...

Fermé

14 mai 2018

#### Historique

#1 - 03 octobre 2017 12:15 - Josué Kouka

- Assigné à changé de Josué Kouka à Paul Marillonnet (retour le 15/04)

## #2 - 03 octobre 2017 12:30 - Paul Marillonnet (retour le 15/04)

"LDAPv3 defines UTF-8 based string encoding for DNs in RFC 4514 and python-ldap happily sends that to the server." (dixit le mainteneur de python-ldap)

Il ne manque pas simplement un str.decode('utf-8') quelque part dans le code du backend LDAP ?

Josué est-ce qu'il y a moyen stp que tu retrouves le DN de l'utilisateur qui a provoqué l'erreur ?  
Je ne vois rien ici ni sur le rapport Sentry.

## #3 - 05 octobre 2017 10:00 - Josué Kouka

Paul Marillonnet a écrit :

"LDAPv3 defines UTF-8 based string encoding for DNs in RFC 4514 and python-ldap happily sends that to the server." (dixit le mainteneur de python-ldap)

Il ne manque pas simplement un str.decode('utf-8') quelque part dans le code du backend LDAP ?

Josué est-ce qu'il y a moyen stp que tu retrouves le DN de l'utilisateur qui a provoqué l'erreur ?  
Je ne vois rien ici ni sur le rapport Sentry.

Tout est dans Sentry u'basedn': u'OU=DSI,OU=DG (Direction générale),OU=Mairie de Dreux,DC=mairiedreux,DC=local'  
Un encode('utf-8') basedn dans le cas où le basedn est une chaîne unicode devrait aller je pense.

## #4 - 05 octobre 2017 15:04 - Paul Marillonnet (retour le 15/04)

worksforme

J'ajoute des entrées utilisateurs de DN non-ascii dans un OpenLDAP (le DN apparaît en base 64 lors d'une commande ldapsearch). La commande sync-ldap-users se passe sans problème, les utilisateurs sont bien ajoutés dans la base authentic2.

## #5 - 05 octobre 2017 15:29 - Thomas Noël

FYI, patch à l'arrache posé par Fred sur la recette authentic/src/authentic2/backends/ldap\_backend.py pour dépanner

```
@@ -319,6 +319,8 @@
        # First get our configuration into a standard format
        for block in blocks:
            cls.update_default(block)
+
+           if isinstance(block.get('basedn'), unicode):
+               block['basedn'] = block['basedn'].encode('utf-8')
            log.debug('got config %r', blocks)
        return blocks
```

## #6 - 06 octobre 2017 16:18 - Benjamin Dauvergne

Si au passage vous pouviez parcourir tous les autres références à du contenu de la config (qui pour mémoire est du JSON et donc ça sort des chaînes unicode), il y aussi user\_basedn et group\_basedn dans ma mémoire qui existent (mais il y a peut-être d'autres setting qui participent à la construction de DNs).

## #7 - 06 octobre 2017 17:01 - Paul Marillonnet (retour le 15/04)

- Fichier 0001-WIP-ldap\_backend-fix-encoding-errors-during-user-syn.patch ajouté
- Patch proposed changé de Non à Oui

Quelque chose comme ça ?

(Je ne vois que ces 4 éléments, je loupe quelque chose ?)

Est-ce que c'est vraiment dans get\_config que cette correction doit avoir lieu ? (Pourquoi pas plutôt directement au bloc le plus local, dans get\_users ?)

## #8 - 09 octobre 2017 10:51 - Benjamin Dauvergne

Hmm effectivement je préfèrerais qu'on ne fasse pas ça. parce que d'autres lib LDAP gère correctement l'unicode, et donc si on y passe à un moment ce sera problématique.

## #9 - 09 octobre 2017 19:15 - Frédéric Péters

(pour info comme il y a eu un niveau paquet authentic dans le dépôt de recette ma modification posée manuellement a été écrasée)

## #10 - 11 octobre 2017 10:46 - Paul Marillonnet (retour le 15/04)

Frédéric, Josué m'a dit avoir ré-appliqué ton patch de façon temporaire en attendant un correctif dans la branche *master*.

Benjamin, est-ce que tu penses qu'il est possible (et préférable) de proposer un patch conservant la même librairie LDAP, en faisant en sorte de ne pas créer de problèmes de compatibilité si jamais on opte pour un changement de librairie ?  
Ou bien est-ce qu'il vaut mieux en profiter pour migrer vers une librairie LDAP qui supporte nativement l'unicode ? (comme le fait ldap3, non ?)

#### #11 - 11 octobre 2017 12:18 - Benjamin Dauvergne

Non on migre rien dans ce ticket, merci. On repère tous les accès à block['basedn'], comme je suis gentil j'ai fait un grep:

```
$ git grep -n block.*dn
src/authentic2/backends/ldap_backend.py:354:             user_basedn = block.get('user_basedn') or block['based
n']
src/authentic2/backends/ldap_backend.py:357:                 if block['user_dn_template']:
src/authentic2/backends/ldap_backend.py:358:                     template = str(block['user_dn_template'])
src/authentic2/backends/ldap_backend.py:525:             group_base_dn = block.get('group_basedn', block['basedn'])
src/authentic2/backends/ldap_backend.py:843:             user_basedn = block.get('user_basedn') or block['based
n']
src/authentic2/backends/ldap_backend.py:952:                 elif block['binddn'] and block['bindpw']:
src/authentic2/backends/ldap_backend.py:953:                     conn.bind_s(block['binddn'], block['bindpw'])
src/authentic2/backends/ldap_backend.py:1075:                         results = conn.search_s(block['basedn
'],
[,
```

et on encode à ces endroits, avec un petit commentaire # python-ldap needs UTF-8 encoded strings.

Possible que des dn soient stockés transitoirement ailleurs (notamment dans la session) faut voir si ça pose problème (les sessions sont stockées en JSON), à la fois en entrée et en sortie (en entrée il me semble que la conversion UTF-8 -> unicode se fait toute seule).

Si on peut avoir un test qui pose des DNs avec des accents c'est cool aussi.

#### #12 - 11 octobre 2017 13:27 - Frédéric Péters

Frédéric, Josué m'a dit avoir ré-appliqué ton patch de façon temporaire en attendant un correctif dans la branche master.

Depuis il y a eu un nouveau paquet authentic poussé vers testing.

#### #13 - 11 octobre 2017 15:14 - Benjamin Dauvergne

Oui c'est moi désolé, mais bon avec 5 personnes qui interviennent sur ce ticket, il y en a bien un qui va me faire un patch :)

#### #14 - 11 octobre 2017 19:06 - Paul Marillonnet (retour le 15/04)

- Fichier 0001-WIP-ldap\_backend-fix-encoding-errors-during-user-syn.patch ajouté

Problème d'environnement de test ce soir, prise de tête, je pose juste un patch "wip" pour indiquer que je suis sur le coup.  
Je soumettrai une version fonctionnelle demain.

#### #15 - 12 octobre 2017 14:07 - Benjamin Dauvergne

Environnement de test ? Marche pas tox ?

#### #16 - 12 octobre 2017 15:49 - Benjamin Dauvergne

(Si c'est pas clair on a un fichier tox.ini à la racine du projet qui permet d'utiliser un outil python nommé tox qui sait construire des virtualenv et lancer les tests, comme ça pas besoin de se souvenir de l'environnement à construire ou de la ligne à la lancer pour tester).

#### #17 - 12 octobre 2017 15:57 - Benjamin Dauvergne

Et j'ai ajouté un peu de doc sur le sujet [HowDoWeDoTests](#).

#### #19 - 17 octobre 2017 16:22 - Paul Marillonnet (retour le 15/04)

- Fichier 0001-WIP-ldap\_backend-fix-encoding-errors-during-user-syn.patch ajouté

Je m'étais engagé à fournir un patch rapidement là-dessus, mais j'ai encore quelques embrouilles avec les tests.

Je suspecte une erreur non détectée à la création du DN dans la fixture ajoutée.  
L'état de mon avancement est dans le patch, je vais continuer à creuser le truc.

#### #20 - 18 octobre 2017 18:04 - Benjamin Dauvergne

Est-ce que le test passe ou pas ? Si non, tu peux recopier la trace ici pour y voir clair.

## #21 - 18 octobre 2017 18:34 - Paul Marillonnet (retour le 15/04)

Non, le test ne passe pas.

Je suspectais une erreur lors de l'ajout de la fixture dans l'annuaire, mais ce n'est pas le cas, comme l'attestent les logs de l'annuaire :

```
59e752b8 conn=1004 fd=16 ACCEPT from PATH=/tmp/a2-provision-slapddK3MYQ/socket (PATH=/tmp/a2-provision-slapddK3MYQ/socket)
59e752b8 conn=1004 op=0 BIND dn="uid=admin,cn=config" method=128
59e752b8 conn=1004 op=0 BIND dn="uid=admin,cn=config" mech=SIMPLE ssf=0
59e752b8 conn=1004 op=0 RESULT tag=97 err=0 text=
59e752b8 conn=1004 op=1 ADD dn="cn=Étienne Michü,o=orga"
59e752b8 conn=1004 op=1 RESULT tag=105 err=0 text=
59e752b8 conn=1004 op=2 ADD dn="cn=group3,o=orga"
59e752b8 conn=1004 op=2 RESULT tag=105 err=0 text=
59e752b8 conn=1004 op=3 UNBIND
59e752b8 conn=1004 fd=16 closed
```

Mais, en dépit du HTTP 200 renvoyé par Authentic, la page HTML contenu dans result correspond à une erreur de login (mire de login, avec champ identifiant prérempli avec la valeur pour laquelle l'authentification n'a pas abouti). Et la base d'utilisateur d'authentic n'est pas provisionnée, User.objects.count() vaut zéro.

La trace en question comme demandée :

```
===== FAILURES =====
_____
test_accents_in_dn _____
```

```
slapd = <ldaptools.slapd.Slapd object at 0x7feae99f0b10>
settings = <pytest_django.fixtures.SettingsWrapper object at 0x7feae4164050>
client = <django.test.client.Client object at 0x7feae4164990>

@ pytest.mark.djangoproject
def test_accents_in_dn(slapd, settings, client):
    settings.LDAP_AUTH_SETTINGS = [
        'url': [slapd.ldap_url],
        'basedn': 'o=entitel',
        'use_tls': False,
    ]
    result = client.post('/login/', {'login-password-submit': '1',
                                    'username': USERNAME_ACCENTS,
                                    'password': PASS}, follow=True)
    assert result.status_code == 200
>       assert u'Étienne Michü' in unicode(str(result), 'utf-8')
\nVary: Cookie, Accept-Language...iv>\n\n \n <script type="text/javascript" src="/static/authentic2/js/js_seconds_until.js"></script>\n\n </body>\n</html>\n'
\nVary: Cookie, Accept-Language...iv>\n\n \n <script type="text/javascript" src="/static/authentic2/js/js_seconds_until.js"></script>\n\n </body>\n</html>\n' = unicode('Content-Language: en\r\nContent-Length: 2938\r\nLast-Modified: Wed, 18 Oct 2017 16:31:23 GMT\r\nVary: Cookie, Accept-... \n <script type="text/javascript" src="/static/authentic2/js/js_seconds_until.js"></script>\n\n </body>\n</html>\n', 'utf-8')
E       +   where 'Content-Language: en\r\nContent-Length: 2938\r\nLast-Modified: Wed, 18 Oct 2017 16:31:23 GMT\r\nVary: Cookie, Accept-... \n <script type="text/javascript" src="/static/authentic2/js/js_seconds_until.js"></script>\n\n </body>\n</html>\n' = str(<django.http.response.HttpResponse object at 0x7feae61e5050>)

tests/test_ldap.py:134: AssertionError
```

## #22 - 18 octobre 2017 21:58 - Benjamin Dauvergne

Ok donc là tu poses un point d'arrêt par exemple dans LdapBackend.authenticate\_block() avec:

```
import pdb
pdb.set_trace()
```

et tu relances ton test:

```
tox -e dj18-authentic-sqlite -- tests/test_ldap.py -k accents --pdb --reuse-db
```

ensuite dans le debugger tu pourrais faire des next et des print pour comprendre ce qui se passe.

Dans le test lui même tu peux utiliser la fixture caplog pour voir les logs.

## #23 - 27 octobre 2017 17:58 - Paul Marillonnet (retour le 15/04)

- Fichier 0001-WIP-ldap\_backend-fix-encoding-errors-during-user-syn.patch ajouté

Ok oui c'est efficace pour voir d'où vient le problème, merci.

Après correction de l'erreur d'incohérence du DN de recherche dans mon précédent pas, j'ai de nouveau une erreur d'encodage.

On dirait qu'une str de filtre LDAP passe au travers des instructions ajoutées ici. J'inspecte la trace :

```
slapd = <ldaptools.slapd.Slapd object at 0x7f416f8e5410>
settings = <pytest_django.fixtures.SettingsWrapper object at 0x7f416f8e5490>

@ pytest.mark.djangoproject_db
def test_nocreate_mandatory_roles(slapd, settings):
    User = get_user_model()
    settings.LDAP_AUTH_SETTINGS = [
        {
            'url': [slapd.ldap_url],
            'basedn': 'o=orga',
            'use_tls': False,
            'create_group': True,
            'group_mapping': [
                ('cn=group2,o=orga', ['Group2']),
            ],
            'group_filter': '(&(memberUid={uid})(objectClass=posixGroup))',
            'set_mandatory_roles': ['tech', 'admin'],
            'create_role': False,
        }
    ]

    users = list(ldap_backend.LDAPBackend.get_users())

tests/test_ldap.py:428:
-----
src/authentic2/backends/ldap_backend.py:869: in get_users
    yield backend._return_user(user_dn, None, conn, block, data)
src/authentic2/backends/ldap_backend.py:815: in _return_user
    return self._return_django_user(dn, username, password, conn, block, attributes)
src/authentic2/backends/ldap_backend.py:826: in _return_django_user
    self.populate_user(user, dn, username, conn, block, attributes)
src/authentic2/backends/ldap_backend.py:638: in populate_user
    self.populate_user_groups(user, dn, conn, block, attributes)
src/authentic2/backends/ldap_backend.py:559: in populate_user_groups
    group_dns = self.get_ldap_group_dns(user, dn, conn, block, attributes)
src/authentic2/backends/ldap_backend.py:550: in get_ldap_group_dns
    results = conn.search_s(group_base_dn, ldap.SCOPE_SUBTREE, query, [])
/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:59
9: in search_s
    return self.search_ext_s(base, scope, filterstr, attrlist, attrsonly, None, None, timeout=self.timeout)
/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:99
8: in search_ext_s
    return self._apply_method_s(SimpleLDAPObject.search_ext_s, *args, **kwargs)
/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:93
6: in _apply_method_s
    return func(self, *args, **kwargs)
/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:59
2: in search_ext_s
    msgid = self.search_ext(base, scope, filterstr, attrlist, attrsonly, serverctrls, clientctrls, timeout, sizelimit)
/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:58
8: in search_ext
    timeout, sizelimit,
-----
self = <ldap.ldapobject.ReconnectLDAPObject instance at 0x7f416e9caeae8>
func = <built-in method search_ext of LDAP object at 0x7f416d48f260>
args = ('o=orga', 2, '(&(memberUid=étienne.michü)(objectClass=posixGroup))', [], 0, None, ...)
kwargs = {}, diagnostic_message_success = None

def _ldap_call(self, func, *args, **kwargs):
    """
        Wrapper method mainly for serializing calls into OpenLDAP libs
        and trace logs
    """
    self._ldap_object_lock.acquire()
    if __debug__:
        if self._trace_level>=1:
            self._trace_file.write('*** %s %s - %s\n%s\n' %
                (repr(self),
                 self._uri,
                 '.'.join((self.__class__.__name__, func.__name__)),
```

```

    pprint.pformat((args,kw_args))
)
if self._trace_level>=9:
    traceback.print_stack(limit=self._trace_stack_limit,file=self._trace_file)
diagnostic_message_success = None
try:
    try:
>       result = func(*args,**kw_args)
E       UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 13: ordinal not in rang
e(128)

/tmp/tox-paul/authentic/coverage-dj18-authentic-sqlite/local/lib/python2.7/site-packages/ldap/ldapobject.py:10
6: UnicodeEncodeError

```

J'ajoute quand même le patch WIP, je vais corriger ça.

#### #24 - 30 octobre 2017 17:00 - Paul Marillonnet (retour le 15/04)

- Fichier *0001-WIP-ldap\_backend-fix-encoding-errors-during-user-syn.patch* ajouté

Compris d'où venait le problème, certaines chaines unicode n'étant pas encodées correctement.

Tous les tests passent, sauf `test_get_users`, le dernier `bulk_create.call_count` vaut 0, et non pas 1 comme le voudrait le test.  
Je regarde d'où ça peut venir, je pose le patch WIP en attendant.

#### #25 - 16 novembre 2017 17:17 - Paul Marillonnet (retour le 15/04)

Je vais déboguer pour voir d'où ça peut venir, mais pour l'instant c'est mystérieux pour moi :

```

=====
 FAILURES =====
_____
 test_get_users _____

slapd = <ldaptools.slapd.Slapd object at 0x7f8b0ad18290>
settings = <pytest_django.fixtures.SettingsWrapper object at 0x7f8b0ad18590>

@ pytest.mark.djangoproject
def test_get_users(slapd, settings):
    import django.db.models.base
    from types import MethodType

    User = get_user_model()
    settings.LDAP_AUTH_SETTINGS = [
        'url': [slapd.ldap_url],
        'basedn': 'o=orga',
        'use_tls': False,
        'create_group': True,
        'group_mapping': [
            ('cn=group2,o=orga', ['Group2']),
        ],
        'group_filter': '(&(memberUid={uid})(objectClass=posixGroup))',
    ]
    save = mock.Mock(wraps=ldap_backend.LDAPUser.save)
    ldap_backend.LDAPUser.save = MethodType(save, None, ldap_backend.LDAPUser)
    bulk_create = mock.Mock(wraps=django.db.models.query.QuerySet.bulk_create)
    django.db.models.query.QuerySet.bulk_create = MethodType(bulk_create, None,
                                                               django.db.models.query.QuerySet)

    # Provision all users and their groups
    assert User.objects.count() == 0
    users = list(ldap_backend.LDAPBackend.get_users())
    assert len(users) == 102
    assert User.objects.count() == 102
    assert bulk_create.call_count == 101
    assert save.call_count == 306

    # Check that if nothing changed no save() is made
    save.reset_mock()
    bulk_create.reset_mock()
    users = list(ldap_backend.LDAPBackend.get_users())
    assert save.call_count == 0
    assert bulk_create.call_count == 0

    # Check that if we delete 1 user, only this user is created
    save.reset_mock()
```

```

bulk_create.reset_mock()
User.objects.last().delete()
assert User.objects.count() == 101
users = list(ldap_backend.LDAPBackend.get_users())
assert len(users) == 102
assert User.objects.count() == 102
assert save.call_count == 3
>   assert bulk_create.call_count == 1
E   AssertionError: assert 0 == 1
E       + where 0 = <Mock id='140235158686544'>.call_count

tests/test_ldap.py:389: AssertionError

```

#### #26 - 16 novembre 2017 18:09 - Paul Marillonnet (retour le 15/04)

Il manque quelque chose AMHA. Je cherchais à bidouiller l'attribut `side_effect` de l'object `bulk_create` "mocké", mais je vais commencer par lire la doc.

#### #27 - 16 novembre 2017 19:44 - Benjamin Dauvergne

Les `bulk_create` viennent de ce code ci :

```

523     def populate_groups_by_mapping(self, user, dn, conn, block, group_dns):
524         '''Assign group to user based on a mapping from group DNs'''
525         group_mapping = block['group_mapping']
526         if not group_mapping:
527             return
528         if not user.pk:
529             user.save()
530             user._changed = False
531         groups = user.groups.all()
532         for dn, group_names in group_mapping:
533             for group_name in group_names:
534                 group = self.get_group_by_name(block, group_name)
535                 if group is None:
536                     continue
537                 # Add missing groups
538                 if dn in group_dns and group not in groups:
539                     user.groups.add(group)
540                 # Remove extra groups
541                 elif dn not in group_dns and group in groups:
542                     user.groups.remove(group)

```

Notamment la ligne 539, faut croire qu'on trouve plus le groupe ou qu'il manque ou que je ne sais quoi.

#### #28 - 23 novembre 2017 17:42 - Paul Marillonnet (retour le 15/04)

Donc tu utilises `bulk_create.call_count` pour compter le nombre de groupes créés lors de la dernière opération de synchro, c'est ça ?

#### #29 - 14 mai 2018 10:56 - Josué Kouka

- Lié à Bug #23698: LDAP: supporter les accents dans les base DNs (et peut-être aussi les filtres, à vérifier) ajouté

#### #30 - 14 mai 2018 11:15 - Benjamin Dauvergne

Ce ticket est un peu différent du [#23698](#).

#### #31 - 04 juillet 2018 16:42 - Benjamin Dauvergne

Bon j'ai bien fait avancer les choses dans le [#23698](#) faudra reprendre par dessus.

#### #32 - 26 octobre 2021 12:06 - Benjamin Dauvergne

- Statut changé de *En cours* à *Rejeté*
- Planning mis à *Non*

On ne voit plus d'erreur d'encodage, je ferme.

### Fichiers

0001-WIP-ldap_backend-fix-encoding-errors-during-user-syn.patch	1,35 ko	06 octobre 2017	Paul Marillonnet (retour le 15/04)
0001-WIP-ldap_backend-fix-encoding-errors-during-user-syn.patch	5,37 ko	11 octobre 2017	Paul Marillonnet (retour le 15/04)
0001-WIP-ldap_backend-fix-encoding-errors-during-user-syn.patch	6,1 ko	17 octobre 2017	Paul Marillonnet (retour le 15/04)

0001-WIP-ldap_backend-fix-encoding-errors-during-user-syn.patch	5,8 ko	27 octobre 2017	Paul Marillonnet (retour le 15/04)
0001-WIP-ldap_backend-fix-encoding-errors-during-user-syn.patch	7,25 ko	30 octobre 2017	Paul Marillonnet (retour le 15/04)