

Idaptools - Development #29418

Fonctionner avec python3

03 janvier 2019 10:42 - Paul Marillonnet

Statut:	Fermé	Début:	03 janvier 2019
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Parce qu'on en aura besoin pour la compatibilité d'A2 avec python3.			

Révisions associées

Révision 7d8c8e94 - 06 mars 2019 15:03 - Paul Marillonnet

add python3 compatibility (#29418)

Historique

#1 - 03 janvier 2019 11:05 - Paul Marillonnet

- Statut changé de Nouveau à Information nécessaire

Étrange, à l'exécution des tests (même en python2) l'instanciation d'un `ldaptools.slapped.Slapped` se fait en allant chercher la clé privée `ssl-cert-snakeoil.key` dans le répertoire de tests `ldaptools` (à savoir, dans mon cas, `~/src/ldaptools/tests/`, et non pas dans `/etc/ssl/`).

Est-ce que je suis censé gérer une nouvelle paire de clés réservée à l'exécution des tests unitaires ?

#2 - 10 janvier 2019 10:08 - Paul Marillonnet

- Statut changé de Information nécessaire à En cours

Le passage en python3 cause des erreurs de parsing LDIF¹, je suis sur le coup.

¹ Par exemple :

```
===== FAILURES =====
_____ test_synchronize_ldif _____
```

```
slapped = <ldaptools.slapped.Slapped object at 0x7fd2b31ed390>
```

```
def test_synchronize_ldif(slapped):
    pivot_attributes = (
        ('organization', 'o'),
        ('inetOrgPerson', 'uid'),
        ('organizationalUnit', 'ou'),
    )
    attributes = ['o', 'objectClass', 'uid', 'sn', 'givenName', 'mail', 'dc', 'cn']
    conn = slapped.get_connection_admin()
```

```
def syn_ldif(ldif):
    parser = ListLDIFParser(StringIO(ldif))
    parser.parse()
    synchronize = Synchronize(parser, 'o=orga', conn, 'o=orga',
                               pivot_attributes=pivot_attributes,
                               attributes=attributes)

    synchronize.run()
    return synchronize
```

```
ldif = '''dn: o=orga
o: orga
dc: coucou
objectClass: organization
objectClass: dcObject
```

```
dn: uid=admin,o=orga
objectClass: inetOrgPerson
cn: John Doe
uid: admin
sn: John
givenName: Doe
mail: john.doe@entrouvert.com
```

```
'''
```

```
> synchronize = syn_ldif(ldif)
```

```
tests/test_synchronize.py:47:
```

```
-----
tests/test_synchronize.py:28: in syn_ldif
    synchronize.run()
src/ldaptools/synchronize.py:285: in run
    self.build_actions()
src/ldaptools/synchronize.py:244: in build_actions
    self.build_actions_for_entries(batch)
src/ldaptools/synchronize.py:187: in build_actions_for_entries
    objectclass, attr, value = self.get_pivot_attribute(dn, entry)
-----
```

```
self = <ldaptools.synchronize.Synchronize object at 0x7fd2b31e35f8>, dn = 'o=orga', entry = caselessDict({'o': [b'orga'], 'dc': [b'coucou'], 'objectclass': <map object at 0x7fd2b32016a0>})
```

```
def get_pivot_attribute(self, dn, entry):
    '''Find a pivot attribute value for an LDAP entry'''
    for objc, attr in self.pivot_attributes:
        entry['objectclass'] = map(istr, entry['objectclass'])
        if objc in entry['objectclass']:
            try:
                value = entry[attr]
            except KeyError:
                raise Exception('entry %s missing pivot attribute %s: %s' % (dn, attr, entry))
            break
    else:
        raise Exception('entry %s has unknown objectclasses %s' % (dn,
            [objclass for objclass in entry['objectclass']]))
>
E      Exception: entry o=orga has unknown objectclasses []
```

```
src/ldaptools/synchronize.py:156: Exception
```

```
test_synchronize_ldap
```

```
slapd = <ldaptools.slapped.Slapped object at 0x7fd2b3239080>
```

```
def test_synchronize_ldap(slapd):
    pivot_attributes = (
        ('organization', 'o'),
        ('inetOrgPerson', 'uid'),
        ('organizationalUnit', 'ou'),
    )
    attributes = ['o', 'objectClass', 'uid', 'sn', 'givenName', 'mail', 'dc', 'cn']
    conn = slapd.get_connection_admin()
```

```
slapd.add_db('dc=orga2')
ldif = '''dn: dc=orga2
o: orga
dc: orga2
objectClass: organization
objectClass: dcObject
```

```
dn: uid=admin,dc=orga2
objectClass: inetOrgPerson
cn: John Doe
uid: admin
sn: John
givenName: Doe
mail: john.doe@entrouvert.com
```

```

'''
    slapd.add_ldif(ldif)

    source = LDAPSource(conn, base_dn='dc=orga2', attributes=attributes)

    synchronize = Synchronize(source, 'dc=orga2', conn, 'o=orga',
                              pivot_attributes=pivot_attributes,
                              attributes=attributes)
>    synchronize.run()

tests/test_synchronize.py:119:
-----
src/ldaptools/synchronize.py:285: in run
    self.build_actions()
src/ldaptools/synchronize.py:244: in build_actions
    self.build_actions_for_entries(batch)
src/ldaptools/synchronize.py:187: in build_actions_for_entries
    objectclass, attr, value = self.get_pivot_attribute(dn, entry)
-----

self = <ldaptools.synchronize.Synchronize object at 0x7fd2b32398d0>, dn = 'dc=orga2', entry = caselessDict({'o': [b'orga'], 'dc': [b'orga2'], 'objectclass': <map object at 0x7fd2b3239d30>})

def get_pivot_attribute(self, dn, entry):
    '''Find a pivot attribute value for an LDAP entry'''
    for objc, attr in self.pivot_attributes:
        entry['objectclass'] = map(istr, entry['objectclass'])
        if objc in entry['objectclass']:
            try:
                value = entry[attr]
            except KeyError:
                raise Exception('entry %s missing pivot attribute %s: %s' % (dn, attr, entry))
            break
    else:
        raise Exception('entry %s has unknown objectclasses %s' % (dn,
            [objclass for objclass in entry['objectclass']]))
>
E      Exception: entry dc=orga2 has unknown objectclasses []

src/ldaptools/synchronize.py:156: Exception

```

#3 - 10 janvier 2019 18:34 - Paul Marillonnet

- Fichier 0001-add-python3-compatibility-29418.patch ajouté
- Statut changé de En cours à Solution proposée
- Patch proposed changé de Non à Oui

#4 - 10 janvier 2019 19:08 - Benjamin Dauvergne

- Fichier 0001-wip.patch ajouté

J'avais ça en local.

#5 - 10 janvier 2019 19:11 - Benjamin Dauvergne

Mais c'est aussi lié à des changements dans python-ldap > 3 qui se met à supporter unicode en python2 (on doit supporter python-ldap 3 même en python2 en fait, il n'y aura plus que ça bientôt).

#6 - 11 janvier 2019 09:46 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Ok cool, merci pour l'info, je vais prendre ça en compte et proposer un nouveau patch, avec tout en python-ldap 3.

Erreur dans mon patch, il y a aussi un

```
if istr(v) in self.objectclasses
```

qui a disparu d'une listcomp dans synchronize.py, je corrige aussi.

#7 - 17 janvier 2019 15:13 - Paul Marillonnet

- Fichier 0001-add-python3-compatibility-29418.patch ajouté

- Statut changé de En cours à Solution proposée

Impossible de faire fonctionner

```
- python-ldap<3
+ ldappre3: python-ldap<3
+ !ldappre3: python-ldap>=3
```

alors que pourtant c'est documenté¹.

Je me mange une erreur de parsing du fichier tox.ini. Pas compris, j'ai dû procéder autrement, rien de bien grave.

Pour le reste c'est ici, avec correction de l'erreur d'étourderie mentionnée dans [#29418-6](#).

¹<https://tox.readthedocs.io/en/latest/config.html#complex-factor-conditions>

#8 - 26 février 2019 23:39 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#9 - 26 février 2019 23:39 - Benjamin Dauvergne

- Fichier 0002-tox.ini-remove-call-to-eobuilder.patch ajouté

- Fichier 0001-add-python3-compatibility-29418.patch ajouté

Relu, poussable.

#10 - 27 février 2019 09:27 - Paul Marillonnet

```
# python2 backward compat
try:
    tested_types = basestring
except NameError:
    tested_types = str
```

Erreur de ma part ici, il suffit d'utiliser six.string_types

#11 - 27 février 2019 09:32 - Paul Marillonnet

- Fichier 0001-add-python3-compatibility-29418.patch ajouté

Corrigé ici.

#12 - 06 mars 2019 13:30 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ack, pousse mon patch eobuilder aussi.

#13 - 06 mars 2019 15:09 - Paul Marillonnet

Est-ce que dans ton patch tu m'autorises aussi à changer :

```
diff --git a/tox.ini b/tox.ini
index 735d177..8fd3386 100644
--- a/tox.ini
+++ b/tox.ini
@@ -5,7 +5,7 @@
```

```
[tox]
toxworkdir = {env:TMPDIR:/tmp}/tox-{env:USER}/ldaptools/
-envlist = py2-coverage-ldap2,py{2,3}-coverage-ldap3,package
+envlist = py2-coverage-ldap2,py{2,3}-coverage-ldap3
```

```
[testenv]
usedevelop = true
```

?

#14 - 06 mars 2019 15:31 - Benjamin Dauvergne

Oui bien vu.

#15 - 06 mars 2019 15:56 - Paul Marillonnet

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 9b7c0e441d4888970cdb9e39aec700c0fe07c1c8
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Tue Feb 26 23:33:09 2019 +0100
```

```
tox.ini: remove call to eobuilder
```

```
commit 7d8c8e9429f253cf2713b14b6f08478369351d31
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date: Thu Jan 10 18:33:25 2019 +0100
```

```
add python3 compatibility (#29418)
```

#16 - 14 mars 2019 22:01 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-python3-compatibility-29418.patch	17,7 ko	10 janvier 2019	Paul Marillonnet
0001-wip.patch	3,05 ko	10 janvier 2019	Benjamin Dauvergne
0001-add-python3-compatibility-29418.patch	18,7 ko	17 janvier 2019	Paul Marillonnet
0002-tox.ini-remove-call-to-eobuilder.patch	846 octets	26 février 2019	Benjamin Dauvergne
0001-add-python3-compatibility-29418.patch	19,3 ko	26 février 2019	Benjamin Dauvergne
0001-add-python3-compatibility-29418.patch	19,2 ko	27 février 2019	Paul Marillonnet