

Abdelmalek Berkani, Haute Ecole Arc, filière informatique de gestion

Oracle Application Express (APEX)

- **Authentification LDAP des utilisateurs de différentes unités organisationnelles (OU)**
- **Rapport interactif basé sur une vue des utilisateurs LDAP**

Authentification LDAP

Oracle APEX permet la création et l'utilisation d'un modèle d'authentification qui permet aux utilisateurs de s'authentifier auprès d'un annuaire LDAP.

L'utilisation d'un annuaire LDAP constitue une bonne pratique «Best practice» en cas de déploiement des applications APEX à destination d'un nombre important d'utilisateurs. L'avantage principal d'un tel modèle est de permettre à l'utilisateur d'utiliser le même nom d'utilisateur et le même mot de passe que ceux qu'ils utilisent déjà pour s'authentifier au système d'exploitation ou à d'autres applications. Côté administrateur, il évite l'administration des comptes à double (noms d'utilisateurs à double, expiration des mots de passe, réinitialisation des mots de passe etc.). La sécurité est également renforcée dans la mesure où si une personne quitte l'entreprise, il suffira de désactiver son compte sur l'annuaire sans devoir le faire dans toutes les applications auxquelles elle accédait.

La configuration du modèle d'authentification LDAP se fait de deux manières dans APEX:

Configuration simple

Tous les utilisateurs LDAP sont dans la même unité organisationnelle (OU) de l'annuaire. Dans ce cas aucune fonction de modification du nom de l'utilisateur LDAP n'est requise.

Exemple: les utilisateurs d'une succursale neuchâteloise d'un domaine «entreprise.ch».

Cette configuration ne peut authentifier que les utilisateurs de la succursale neuchâteloise. Un utilisateur en dehors de cette unité organisationnelle ne pourra pas se connecter.

Configuration avancée

Les utilisateurs LDAP sont dans des unités organisationnelles différentes. Une fonction de modification du nom de l'utilisateur est nécessaire.

Exemples:

La succursale de Neuchâtel:
OU=Neuchatel,DC=entreprise,DC=ch

La succursale de Zurich:
OU=Zurich,DC=entreprise,DC=ch

En partant du postulat qu'il n'est pas toujours possible ou souhaitable de mettre tous les utilisateurs à l'intérieur d'une même unité organisationnelle, il faut trouver une solution pour permettre aux utilisateurs des deux succursales de s'authentifier.

La solution consiste à:

1. Créer un utilisateur générique dans l'annuaire LDAP dans une unité or-

ganisationnelle connue et stable et ayant un mot de passe qui n'expire pas.

Exemple: CN=ldapUser,OU=Utilisateurs,DC=entreprise,DC=ch

2. Créer une fonction PL/SQL qui se connecte à l'annuaire LDAP avec cet utilisateur et qui retourne la «distinguishedName» de l'utilisateur qui veut se connecter à APEX. Veuillez consulter le code de la fonction «GET_DN» du package AUT_LDAP en annexe.
3. Configurer le modèle d'authentification dans APEX.

[Intégré] [Compte de base de données] [LDAP]
 Hôte LDAP [Outil de test LDAP]
 ldapserv.entreprise.ch
 Port LDAP
 389
 Chaîne DN LDAP
 %LDAP_USER%
 Fonction de modification de nom utilisateur LDAP
 return aut_ldap.get_dn

Cette configuration fonctionne de la manière suivante:

- L'utilisateur saisit son nom d'utilisateur et son mot de passe sur la page de connexion APEX.
- APEX passe le nom de l'utilisateur saisi à la fonction GET_DN. Veuillez noter que les fonctions de modification du nom utilisateur LDAP doivent disposer d'un argument p_username qu'APEX utilise de manière implicite, c'est pour cette raison que l'appel de la fonction se fait sans cet argument.
- La fonction GET_DN se connecte à l'annuaire en utilisant un utilisateur générique, elle parcourt l'annuaire et dès qu'elle trouve le bon utilisateur, elle retourne sa «distinguishedName».
- APEX remplace %LDAP_USER% par la «distinguishedName» retournée par la fonction GET_DN et demande la connexion au serveur LDAP.

Rapport interactif basé sur LDAP

L'idée est de pouvoir consulter depuis APEX la liste des utilisateurs LDAP en direct et sans passer par aucune table de la base de données.

La solution consiste à:

1. Créer un type objet
Le type «LDAP_LIGNE» dans l'annexe.
2. Créer une Nested Table
Le type «LDAP_TABLE» dans l'annexe.
3. Créer une fonction
La fonction «GET_LDAP_USERS» dans l'annexe.
4. Créer une vue
La vue «V_LDAP» dans l'annexe.
5. Utiliser la vue comme source SQL dans un rapport interactif.

```
SELECT username, date_creation „Date Création“, date_modif „Date Modification“
FROM v_ldap
```

Utilisateurs LDAP

Lignes 15

Nom Utilisateur	Date Création	Date Modification
zbggglwd	18.01.2008	05.12.2008
zckwrmzn	19.09.2008	05.12.2008
zwbwrlls	05.09.2007	28.01.2009
zwwgrrwr	13.07.2004	28.01.2009
zwlwnll	30.01.2009	27.02.2009
zffflkwj	10.08.2005	05.12.2008
zgrwbis	30.01.2009	27.02.2009

Annexes:

Type objet et Nested Table:

```
CREATE OR REPLACE TYPE LDAP_LIGNE
AS OBJECT (USERNAME      VARCHAR2(50),
           NOM           VARCHAR2(100),
           PRENOM       VARCHAR2(100),
           MAIL         VARCHAR2(100),
           TEL          VARCHAR2(50),
           DN           VARCHAR2(255),
           DATE_CREATION DATE,
           DATE_MODIF   DATE)
/
CREATE OR REPLACE TYPE LDAP_TABLE IS TABLE OF LDAP_LIGNE
/
```

Package AUT_LDAP, fonctions GET_DN et GET_LDAP_USERS

```
CREATE OR REPLACE PACKAGE AUT_LDAP AS
  co_ldap_host   VARCHAR2(256) := ,ldapsver.entreprise.ch';
  co_ldap_port   VARCHAR2(256) := ,389';
  co_ldap_user   VARCHAR2(256) := ,CN=ldapUser,OU=Utilisateurs,DC=entreprise,DC=ch';
  co_ldap_pwd    VARCHAR2(256) := ,ldapPassword';
  co_ldap_base   VARCHAR2(256) := ,DC=entreprise,DC=ch';

  Function get_dn(p_username in varchar2) return varchar2;
  Function get_ldap_users return LDAP_TABLE;
END AUT_LDAP;
/

CREATE OR REPLACE PACKAGE BODY AUT_LDAP AS

  Function get_dn(p_username in varchar2)
  return varchar2
  AS
    v_retval      PLS_INTEGER;
    v_session     DBMS_LDAP.session;
    v_attrs       DBMS_LDAP.string_collection;
    v_message     DBMS_LDAP.message;
    v_entry       DBMS_LDAP.message;
    v_attr_name   VARCHAR2(256);
    v_ber_element DBMS_LDAP.ber_element;
    v_vals        DBMS_LDAP.string_collection;
    v_Retour      Varchar2(255);
    v_Filter      Varchar2(255);

  BEGIN
    v_Retour := p_username;
    --On restreint le filtre pour éviter la limite de retour: 1000 entrées max
    v_Filter := '(&(sAMAccountName= | | p_username | | )(objectclass=user))';

    --Exceptions
    DBMS_LDAP.USE_EXCEPTION := TRUE;

    -- Connexion
    v_session := DBMS_LDAP.init(hostname => AUT_LDAP.co_ldap_host,
                                portnum => AUT_LDAP.co_ldap_port);

    v_retval := DBMS_LDAP.simple_bind_s(ld      => v_session,
                                        dn      => AUT_LDAP.co_ldap_user,
                                        passwd => AUT_LDAP.co_ldap_pwd);

    -- On ne s'intéresse qu'à la dn
    v_attrs(1) := 'distinguishedName';
    v_retval := DBMS_LDAP.search_s(ld      => v_session,
                                   base    => AUT_LDAP.co_ldap_base,
                                   scope   => DBMS_LDAP.SCOPE_SUBTREE,
                                   filter  => v_Filter,
                                   attrs   => v_attrs,
                                   attronly => 0,
                                   res     => v_message);
```

```

IF DBMS_LDAP.count_entries(ld => v_session, msg => v_message) > 0 THEN
-- Toutes les entrées
v_entry := DBMS_LDAP.first_entry(ld => v_session,
                                msg => v_message);

WHILE v_entry IS NOT NULL LOOP
-- Tous les attributs de cette entrée
v_attr_name := DBMS_LDAP.first_attribute(ld => v_session,
                                         ldapentry => v_entry,
                                         ber_elem => v_ber_element);

-- attributes loop
WHILE v_attr_name IS NOT NULL LOOP
-- Toutes les valeurs de cet attribut
v_vals := DBMS_LDAP.get_values (ld => v_session,
                                ldapentry => v_entry,
                                attr => v_attr_name);

FOR i IN v_vals.FIRST .. v_vals.LAST LOOP
if instr(upper(v_vals(i)),upper(p_username)) > 0 then
v_Retour := v_vals(i);
exit;

end if
;

END LOOP;
v_attr_name := DBMS_LDAP.next_attribute(ld => v_session,
                                       ldapentry => v_entry,
                                       ber_elem => v_ber_element);

END LOOP attributes_loop;
v_entry := DBMS_LDAP.next_entry(ld => v_session,
                                msg => v_entry);

END LOOP;
END IF;

-- Déconnexion
v_retval := DBMS_LDAP.unbind_s(ld => v_session);
return v_Retour;
END get_dn;

Function GET_LDAP_USERS return LDAP_TABLE
is
co_filter          VARCHAR2(100) := 'objectclass=user';

LDAP_USERS LDAP_TABLE := LDAP_TABLE(LDAP_LIGNE(NULL, NULL,NULL, NULL,
                                               NULL, NULL,NULL, NULL));

retval            PLS_INTEGER;
v_session         DBMS_LDAP.session;
v_attrs           DBMS_LDAP.string_collection;
v_message         DBMS_LDAP.message;
v_entry           DBMS_LDAP.message;
v_dn              VARCHAR2(256);
v_attr_name       VARCHAR2(256);
v_ber_elmt        DBMS_LDAP.ber_element;

i                 PLS_INTEGER;
v_vals            DBMS_LDAP.STRING_COLLECTION ;
b_first           boolean := TRUE;

v_username        varchar2(50);
v_nom             varchar2(100);
v_prenom          varchar2(100);
v_mail            varchar2(100);
v_tel             varchar2(50);
v_date_creation   date;
v_date_modif      date;

BEGIN
retval            := -1;

-- Exceptions
DBMS_LDAP.USE_EXCEPTION := TRUE;

--Connexion
v_session := DBMS_LDAP.init(AUT_LDAP.co_ldap_host,AUT_LDAP.co_ldap_port);
retval := DBMS_LDAP.simple_bind_s(v_session, AUT_LDAP.co_ldap_user, AUT_LDAP.co_ldap_pwd);

```

```

--Limiter la recherche aux attributs LDAP suivants
--Username
v_attrs(1) := 'sAMAccountName';
--nom
v_attrs(2) := 'sn';
--prenom
v_attrs(3) := 'givenName';
--mail
v_attrs(4) := 'mail';
--tel
v_attrs(5) := 'telephoneNumber';
--dn
v_attrs(6) := 'distinguishedName';
--date creation
v_attrs(7) := 'whenCreated';
-- date modif
v_attrs(8) := 'whenChanged';

--recherche
retval := DBMS_LDAP.search_s(v_session, co_ldap_base,
                           DBMS_LDAP.SCOPE_SUBTREE,
                           co_filter,
                           v_attrs,
                           0,
                           v_message);

-- Compter le nombres des entrées LDAP
retval := DBMS_LDAP.count_entries(v_session, v_message);

-- La première entrée
v_entry := DBMS_LDAP.first_entry(v_session, v_message);

-- Boucle des entrées LDAP
while v_entry IS NOT NULL
loop

    v_attr_name := DBMS_LDAP.first_attribute(v_session,v_entry,v_ber_elmt);

    --Mettre les valeurs pour éviter qu'un user qui n'a pas un attribut ldap
    --comme telephoneNumber ait la valeur de l'attribut du dernier user l'ayant eu.
    v_username := null;
    v_nom := null;
    v_prenom := null;
    v_mail := null;
    v_tel := null;
    v_dn := null;
    v_date_creation := null;
    v_date_modif := null;

    --Boucle des attributs
    while v_attr_name IS NOT NULL
    loop
        v_vals := DBMS_LDAP.get_values(v_session,v_entry,v_attr_name);
        if v_vals.COUNT > 0
        then
            --Boucles des valeurs par attribut
            FOR i in v_vals.FIRST..v_vals.LAST
            loop
                CASE v_attr_name
                WHEN v_attrs(1) THEN
                    v_username := substr(v_vals(i),1,50);
                WHEN v_attrs(2) THEN
                    v_nom := substr(v_vals(i),1,100);
                WHEN v_attrs(3) THEN
                    v_prenom := substr(v_vals(i),1,100);
                WHEN v_attrs(4) THEN
                    v_mail := substr(v_vals(i),1,100);
                WHEN v_attrs(5) THEN
                    v_tel := substr(v_vals(i),1,50);
                WHEN v_attrs(6) THEN
                    v_dn := substr(v_vals(i),1,255);
                WHEN v_attrs(7) THEN
                    v_date_creation := to_date(substr(v_vals(0),1,14), 'yyyymmddhh24miss');
                WHEN v_attrs(8) THEN
                    v_date_modif := to_date(substr(v_vals(0),1,14), 'yyyymmddhh24miss');
                END case;
            end loop;
        end if;
        v_attr_name := DBMS_LDAP.next_attribute(v_session,v_entry,v_ber_elmt);
    end loop;
end loop;

```

```

if b_first
then
  b_first := FALSE;
else
  LDAP_USERS.extend;
end if;

LDAP_USERS(LDAP_USERS.last) := LDAP_LIGNE(v_username,
                                          v_nom,
                                          v_prenom,
                                          v_mail,
                                          v_tel,
                                          v_dn,
                                          v_date_creation,
                                          v_date_modif);

v_entry := DBMS_LDAP.next_entry(v_session, v_entry);
end loop;
-- Déconnexion LDAP
retval := DBMS_LDAP.unbind_s(v_session);
return(LDAP_USERS);
END GET_LDAP_USERS;

END AUT_LDAP;
/

```

La vue V_LDAP

```

CREATE OR REPLACE VIEW V_LDAP AS
(
  select USERNAME, NOM, PRENOM, MAIL, TEL, DN, DATE_CREATION, DATE_MODIF
  from Table( Cast( AUT_LDAP.GET_LDAP_USERS() As LDAP_TABLE ));
/

```

Contact

Haute Ecole Arc,
 filière informatique de gestion

Abdelmalek Berkani
 E-Mail: Abdelmalek.Berkani@he-arc.ch

SMS

Oracle Delivers World Record Java Virtual Machine Performance for an x86-Based System on Linux with SPECjbb2005 Benchmark

Oracle announced that Oracle® JRockit, Java Virtual Machine (JVM), achieved a world-record performance for an x86-based system running Linux on the SPECjbb2005 benchmark, an industry-standard measurement of server side Java-based application performance.

Oracle JRockit delivered 2,150,260 SPECjbb2005 bops, running on a 16-socket NEC Express5800/

A1160 with a six-core Intel® Xeon® X7460 2.67 GHz processor and 256 GB RAM on Linux.

The result delivered more than two times higher performance than the top SPECjbb2005 benchmark results on x86 using Sun JVM and nearly three times higher performance than IBM JVM. Additionally, this Oracle JRockit result outperforms similarly sized RISC systems based on Sun JVM running on Sun

SPARC by 163 percent and IBM JVM running on IBM POWER by 40 percent.

Oracle JRockit integrates technology from Oracle Fusion Middleware and BEA Systems and illustrates the rapid progress that Oracle is making in combining industry-leading technologies from the two companies into a unified product offering.