# Avoiding and resetting expired passwords in Oracle databases

July 29, 2014 by [Ingolf Loboda](http://www.ateam-oracle.com/author/ingolf-loboda/) [1 Comment](http://www.ateam-oracle.com/avoiding-and-resetting-expired-passwords-in-oracle-databases/" \l "comments)

# **Introduction**

Doing a default database installation will install a feature that all passwords will expire after 180 days.

Not being aware of that can cause problems in applications as they cannot connect to the database after that time period. Especially if you are working in test or development environment you will mostly not care about security concerns. That will mean that there is no real need that passwords should expire automatically.

The purpose of this post is to show how the behavior could be changed and how expired accounts can be reset even if they are expired.

# Main Article

## Adjusting the password expiration policy

First we have to verify the password expiration policy in your database. The best way to do that is running a query against DBA\_PROFILES:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| SQL> select \* from dba\_profiles where resource\_name = 'PASSWORD\_LIFE\_TIME';    PROFILE               RESOURCE\_NAME                   RESOURCE LIMIT  ----------------      --------------------------      -----------------------  DEFAULT               PASSWORD\_LIFE\_TIME              PASSWORD 180 |

This confirms the password expiry policy is set to the default of 180 days. You can change this value to unlimited in the following way:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| ALTER PROFILE DEFAULT LIMIT PASSWORD\_LIFE\_TIME UNLIMITED; |

Keep in mind to adjust the statement if you are using a different profile name as ‘DEFAULT’ or multiple profiles.

The verification of this change is done using the same query as before:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| SQL> select \* from dba\_profiles where resource\_name = 'PASSWORD\_LIFE\_TIME';    PROFILE               RESOURCE\_NAME                   RESOURCE LIMIT  ----------------      --------------------------      -----------------------  DEFAULT               PASSWORD\_LIFE\_TIME              PASSWORD UNLIMITED |

## Checking for problematic accounts

What options do you have in case you mentioned that problem too late and you see users which are marked as “EXPIRED” or even marked as “LOCKED”?

The sql statement listed now will give you an overview if there are any expired or locked accounts visible in your database.

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| SQL> select username,account\_status from dba\_users where account\_status like '%EXPIRED%'  or account\_status like '%LOCKED%';    USERNAME      ACCOUNT\_STATUS  ----------    --------------------  MDSYS         EXPIRED & LOCKED  ORACLE\_OCM    EXPIRED & LOCKED  ORDDATA       EXPIRED & LOCKED  ORDPLUGINS    EXPIRED  ORDSYS        EXPIRED & LOCKED  OUTLN         LOCKED |

#### *Resetting of problematic accounts*

The next sections will show two different options how to reopen the affected accounts using scripts. Both version will reopen the accounts so that you can work with them again.

The main difference between both options is that option 1 will set all expired accounts to a new password given in the script.

Option 2 will do the same but reset the account passwords to the old password instead of setting new passwords.

You should choose the option which will fit the best for your needs.

#### *Option 1: Assign a new password*

You need to create a sql command file named : userpwn.sql (the name is only an example and free to use).

This needs to have the following lines:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| spool on;  set echo off;  set heading off;  set feedback off;  SET   SERVEROUTPUT  OFF;  spool unlock.sql;  select 'ALTER USER '|| USERNAME || ' account unlock;' from dba\_users where ACCOUNT\_STATUS like '%LOCKED%';  spool off;  @unlock.sql;  spool on;  set echo off;  set heading off;  set feedback off;  SET   SERVEROUTPUT  OFF;  spool pwchangen.sql;  select 'ALTER USER '|| USERNAME || ' identified by password1;' from dba\_users  where ACCOUNT\_STATUS like '%EXPIRED%' or ACCOUNT\_STATUS like '%LOCKED%';  spool off;  @pwchangen.sql; |

Keep in mind. All expired accounts will get the new password “password1” in this example. You can change the value according to your needs.

#### *Option 2: restore previous password*

You need to create a sql command file named : userpwn.sql (the name is only an example and free to use).

This needs to have the following lines:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| spool on;  set echo off;  set heading off;  set feedback off;  SET   SERVEROUTPUT  OFF;  spool unlock.sql;  select 'ALTER USER '|| USERNAME || ' account unlock;' from dba\_users where ACCOUNT\_STATUS like '%LOCKED%';  spool off;  @unlock.sql;  spool on;  set lines 300;  set echo off;  set heading off;  set feedback off;  SET   SERVEROUTPUT  OFF;  spool pwchangeo.sql;  select 'ALTER USER '|| USERNAME || ' identified by values ''' || spare4 || ''';' from dba\_users,user$  where ACCOUNT\_STATUS like '%EXPIRED%' and USERNAME=NAME;  spool off;  @pwchangeo.sql; |

In this version all expired accounts will get the same password as before. So any client which is connecting to the database will work again without doing any changes on that side.

## Examples of scripts used

For a better understanding we will now show examples of the sql statements generated using the different options.

The sql statements generated for unlock.sql will look like :

[?](http://www.ateam-oracle.com/)

|  |
| --- |
|  |

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| ALTER USER MDSYS account unlock;  ALTER USER ORDSYS account unlock;  ALTER USER EXFSYS account unlock;  ALTER USER DBSNMP account unlock; |

An example of the statements for pwchangen.sql is listed here:

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| ALTER USER MDSYS identified by password1;  ALTER USER ORDSYS identified by password1;  ALTER USER EXFSYS identified by password1;  ALTER USER DBSNMP identified by password1; |

And last here is an example of generated statements for pwchangeo.sql :

[?](http://www.ateam-oracle.com/)

|  |
| --- |
| ALTER USER MDSYS identified by values 'S:CC17B63ECCC75D78911698A779C37AC7713C7DA268D3E3B0DFF36BCEF659';  ALTER USER ORDSYS identified by values 'S:7F482CD28FBF61B255A2950C16C94F8F7B36185F5CA64FDB336985A99A36';  ALTER USER EXFSYS identified by values 'S:D94E9A865A3CFBCBCBB57D5146071E4B113B31C4077D81F291CBF2873E86';  ALTER USER DBSNMP identified by values 'S:DD9AD8BCCF9BE89A85056DD157DEFAC9F3EAC4D36F33F3609783F8C7B472'; |

All site content is the property of Oracle Corp. Redistribution not allowed without written permission

Filed Under: [Fusion Applications](http://www.ateam-oracle.com/category/fusion-apps/), [Install and Provision](http://www.ateam-oracle.com/category/fusion-apps/fa-install/)

### Comments

IMG_256

[Andy Knight](http://www.ateam-oracle.com/author/andy-knight/) says:

[October 5, 2018 at 1:21 AM](http://www.ateam-oracle.com/avoiding-and-resetting-expired-passwords-in-oracle-databases/" \l "comment-4244)

As of 11.2.0.4 and 12c, it is not possible to use “by values” as this was always meant for internal (data pump) usage only. This has now been enforced.

Therefore “Option 2: restore previous password” will not work in 12c of 11.2.0.4 (or higher)

[Log in to Reply](http://www.ateam-oracle.com/wp-login.php?redirect_to=http://www.ateam-oracle.com/avoiding-and-resetting-expired-passwords-in-oracle-databases/)

### Add Your Comment [Cancel reply](http://www.ateam-oracle.com/avoiding-and-resetting-expired-passwords-in-oracle-databases/" \l "respond)

You must be [logged in](http://www.ateam-oracle.com/wp-login.php?redirect_to=http://www.ateam-oracle.com/avoiding-and-resetting-expired-passwords-in-oracle-databases/) to post a comment.