Password Policy Configuration

This document describes the configuration of a password policy for DigDash Enterprise.

The password policy configuration is separated in two parts:

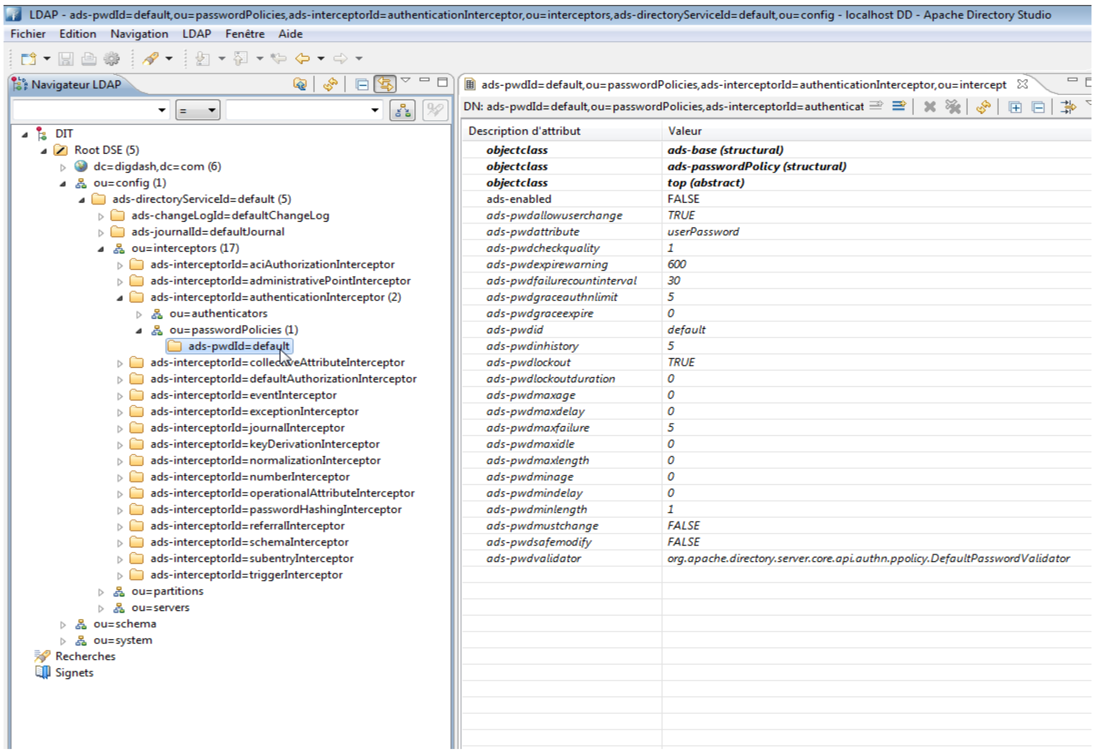
- The strategies of **protection**, **life cycle**, and a part of the password **quality** are defined directly in the LDAP server natively supported by DigDash Enterprise (OpenLDAP 2.0+)
- The password **quality** rules (syntax) are defined in a configuration file in DigDash Enterprise.

For the LDAP configuration part, we recommend the use of the software **Apache Directory Studio**.

I. ACCESSING LDAP PASSWORD POLICY

The password policy in LDAP is enabled through an LDAP interceptor (enabled by default). To configure it you can connect to the DigDash Enterprise LDAP with **Apache Directory Studio**, with an LDAP administrator user.
Once connected, expand the following nodes (see screenshot for more details):

```
ou=config
        ads-directoryServiceId=default
                ou=interceptors
                        ads-interceptorId=authenticationInterceptor
                                ou=passwordPolicies
                                        ads-pwdId=default
```



LDAP - ads-pwdId=default,ou=passwordPolicies,ads-interceptorId=authenticationInterceptor,ou=interceptors,ads-directoryServiceId=default,ou=config - localhost DD - Apache Directory Studio

Fichier Edition Navigation LDAP Fenêtre Aide

Navigateur LDAP

```
DIT
  Root DSE (5)
    dc=digdash,dc=com (6)
    ou=config (1)
      ads-directoryServiceId=default (5)
        ads-changeLogId=defaultChangeLog
        ads-journalId=defaultJournal
        ou=interceptors (17)
          ads-interceptorId=aciAuthorizationInterceptor
          ads-interceptorId=administrativePointInterceptor
          ads-interceptorId=authenticationInterceptor (2)
            ou=authenticators
            ou=passwordPolicies (1)
              ads-pwdId=default
          ads-interceptorId=collectiveAttributeInterceptor
          ads-interceptorId=defaultAuthorizationInterceptor
          ads-interceptorId=eventInterceptor
          ads-interceptorId=exceptionInterceptor
          ads-interceptorId=journalInterceptor
          ads-interceptorId=keyDerivationInterceptor
          ads-interceptorId=normalizationInterceptor
          ads-interceptorId=numberInterceptor
          ads-interceptorId=operationalAttributeInterceptor
          ads-interceptorId=passwordHashingInterceptor
          ads-interceptorId=referralInterceptor
          ads-interceptorId=schemaInterceptor
          ads-interceptorId=subentryInterceptor
          ads-interceptorId=triggerInterceptor
        ou=partitions
        ou=servers
    ou=schema
    ou=system
Recherches
Signets
```

ads-pwdId=default,ou=passwordPolicies,ads-interceptorId=authenticationInterceptor,ou=intercept

DN: ads-pwdId=default,ou=passwordPolicies,ads-interceptorId=authenticat

Description d'attribut	Valeur
objectclass	***ads-base (structural)***
objectclass	***ads-passwordPolicy (structural)***
objectclass	***top (abstract)***
ads-enabled	FALSE
ads-pwdallowuserchange	*TRUE*
ads-pwdattribute	*userPassword*
ads-pwdcheckquality	*1*
ads-pwdexpirewarning	*600*
ads-pwdfailurecountinterval	*30*
ads-pwdgraceauthnlimit	*5*
ads-pwdgraceexpire	*0*
ads-pwdid	*default*
ads-pwdinhistory	*5*
ads-pwdlockout	*TRUE*
ads-pwdlockoutduration	*0*
ads-pwdmaxage	*0*
ads-pwdmaxdelay	*0*
ads-pwdmaxfailure	*5*
ads-pwdmaxidle	*0*
ads-pwdmaxlength	*0*
ads-pwdminage	*0*
ads-pwdmindelay	*0*
ads-pwdminlength	*1*
ads-pwdmustchange	*FALSE*
ads-pwdsafemodify	*FALSE*
ads-pwdvalidator	*org.apache.directory.server.core.api.authn.ppolicy.DefaultPasswordValidator*

I.1 Main attributes description

The following table describes the main attributes used to configure the password policy in OpenLDAP:

Function	**Attribute**	**Type**	**Default value**	**Description**
Protection	Password protection against attacks. Lock the password after a certain number of unsuccessful attempts to log in occur.			
	ads-pwdlockout	Boolean	TRUE	**TRUE**: Enable password lock **FALSE**: disable password lock

	ads-pwdlockoutduration	Integer (seconds)	0	Defines the duration of the password lock. **0**: infinite duration: An LDAP admin must unlock the password of the account.
	ads-pwdfailurecountinterval	Integer (seconds)	30	Defines the delay before the unsuccessful log in attempts counter is reset.
	ads-pwdmaxfailure	Integer	5	If the password lock is enabled (ads-pwdlockout = TRUE), this attribute defines the number of log in failures that will lock the password.
Quality	Password quality rules *Note: By default DigDash Enterprise stores hashed passwords in LDAP. So LDAP does not know the original password entered by the user, and so can not check the its quality. This check is done directly within DigDash Enterprise (see following chapter in this document). The following attributes are documented for the case when the default behaviour would be modified. However, some of them are used (eg. history).*			
	ads-pwdcheckquality	Integer	1	Type of password quality: **0**: Password quality is not checked **1**: Password quality is checked when it is possible (not hashed). If the password is hashed, or in a form impossible to check, the password is accepted. **2**: Password quality is always checked. If the password is hashed, or in a form impossible to check, the password is rejected. *Note: Password quality check is done directly within DigDash Enterprise. You must leave this attribute to 0 or 1.*
	ads-pwdinhistory	Integer	5	LDAP can keep a history of previous passwords for a user. This attribute defines the number of history entries.
	ads-pwdminage	Integer (seconds)	0	If the password history is enabled this attribute defines the minimal delay between two successive password changes.
	ads-pwdminlength	Integer (characters)	1	This attribute defines the minimum length of the password in characters. *Note: Password quality check is done directly within DigDash Enterprise. You must leave this attribute to 1*

	ads-pwdmaxlength	Integer (characters)	0	This attribute defines the maximum length of the password in characters. *Note: Password quality check is done directly within DigDash Enterprise. You must leave this attribute to 0.*
Life Cycle	Password life cycle management.			
	ads-pwdallowuserchange	Boolean	TRUE	**TRUE**: The user can change its own password. **FALSE**: The user can not change its own password.
	ads-pwdexpirewarning	Integer (seconds)	600	Defines if the LDAP server should answer with a warning when a password is about to expire in the specified delay.
	ads-pwdgraceauthnlimit	Integer	5	When the password expires, this attribute defines the number of times the password is still usable before its definitive expiration. *Note: DigDash Enterprise consumes one token of this counter to change the password. You should add 1 to the specified value regarding the real number of grace limit you want to specify: If ads-pwdgraceauthnlimit = 6, it means there are 5 logins allowed before expiration of the password. The warning messages take this offset into account.*
	ads-pwdgraceexpire	Integer (seconds)	0	When the password expires, this attribute defines the remaining period of time when the password is still usable before its definitive expiration.. **0**: no grace period
	ads-pwdmaxage	Integer (seconds)	0	Maximum age of the password before expiration. **0**: password never expires
	ads-pwdmaxidle	Integer (seconds)	0	Maximum idle time duration of the password. The password expires when this time is over.
	ads-pwdmustchange	Boolean	FALSE	**TRUE**: The password must be changed
	ads-pwdsafemodify	Boolean	FALSE	For password change. **TRUE**: The user must enter his previous password to change it. **FALSE**: The user does not have to enter his previous password to change it.

				Note: DigDash Enterprise already forces the user to enter its previous password to change it. This parameter must be set to FALSE.
Other	Other advanced functions and/or not supported.			
	ads-pwdmindelay	Integer	0	Not supported
	ads-pwdmaxdelay	Integer	0	Not supported
	ads-pwdattribute	String	userPassword	Name of the attribute where the password is stored in LDAP. *Note: Changing this parameter is not recommened in DigDash Enterprise.*

Note: Changes of the password policy parameters are applied on the new passwords. Existing passwords keep the password policy that was in place when they were created.

Important: Version 2018R2 introduces the support for more secure SHA hashing algorithms than default SHA-1 for LDAP. Using one of the "salted" SHA algorithm (eg. SSHA…) can interfere with some feature of the password policy, like the password history constraint (ads-pwdinhistory). Actually, the salt is based on a random number to ensure that compromising one password would not compromise all stored hashed passwords. It also helps to limit effectiveness of brute-force dictionary attacks on the hashes. But that random number prevents LDAP from comparing a new password to the old ones in the password history. If this feature is mandatory, then we recommend using a not salted SHA algorithm like SHA-512 and a strong password.

II. PASSWORD QUALITY CONFIGURATION (DIGDASH ENTERPRISE SPECIFIC)

By default DigDash Enterprise stores hashed passwords in LDAP. So LDAP does not know the original password entered by the user, and so can not check the its quality. This check is done directly within DigDash Enterprise.

This chapter describes how to specify constraint on password within DigDash Enterprise.

Password quality rules are defined in the **passwordpolicyrepository.xml** file.

A default file is delivered with DigDash Enterprise, but does not define any constraint on user passwords. The default configuration file is located in the ddenterpriseapi web application folder but it is not recommended to directly modify it at this location, unless loosing the modification in a future DigDash Enterprise upgrade.

To modify the password quality rules the simplest procedure is:

1. **Copy** the default file located at the following place:
   <DDE Install>/apache-tomcat/webapps/**ddenterpriseapi**/WEB-INF/classes/resources/config/**passwordpolicyrepository.xml**
   to the following place:
   <user>/Application Data/Enterprise Server/ddenterpriseapi/config/**passwordpolicyrepository.xml**
2. **Modify** the copy with a text editor
3. **Restart** the Tomcat server after the modification.

*II.1 File format of **passwordpolicyrepository.xml***

The default file contains the following XML:

```
<PasswordPolicyRepository>
      <Rules>
            <Profil>administrator</Profil>
            <Pattern>.*</Pattern>
            <MustHaveUpperCase>false</MustHaveUpperCase>
            <MustHaveLowerCase>false</MustHaveLowerCase>
            <MustHaveNumeric>false</MustHaveNumeric>
            <MustHaveSpecialChar>false</MustHaveSpecialChar>
            <MustNotContainID>false</MustNotContainID>
      </Rules>
      <Rules>
            <Profil>user</Profil>
            <Pattern>.*</Pattern>
            <MustHaveUpperCase>false</MustHaveUpperCase>
            <MustHaveLowerCase>false</MustHaveLowerCase>
            <MustHaveNumeric>false</MustHaveNumeric>
            <MustHaveSpecialChar>false</MustHaveSpecialChar>
            <MustNotContainID>false</MustNotContainID>
      </Rules>
</PasswordPolicyRepository>
```

It defines two rules, one for the DigDash Enterprise administrator profile (admin...), and the other for the profile of the other users. The two rules have the same syntax.

These two profiles will allow to specify a different password quality for the administrators and the regular users.

II.2 Rule parameters

Parameter	**Type**	**Default value**	**Description**
Profil	String	administrator user	Name of the user profile concerned by this rule: **administrator**: the rule applies to the DigDash Enterprise administrators (admin...) **user**: the rule applies to all other regular DigDash Enterprise users. *Note: At this time no other value will be accepted by DigDash Enterprise.*
Pattern	String (regular expression)	.*	Optional regular expression to allow a more complex password syntax constraint, in addition to the other parameters (See next chapter). **.***: any syntax allowed. If the password doe not comply to the regular expression, it is rejected whatever its compliance with the other parameter of the rule.
MustHaveUpperCase	Boolean	false	**true**: password must contain at least one upper-case letter. **false**: password may contain upper-case letters or not.
MustHaveLowerCase	Boolean	false	**true**: password must contain at least one lower-case letter. **false**: password may contain lower-case letters or not.
MustHaveNumeric	Boolean	false	**true**: password must contain at least one digit. **false**: password may contain digits or not.
MustHaveSpecialChar	Boolean	false	**true**: password must contain at least one character which is not a letter or a digit. **false**: password may contain only letters and/or digits.
MustNotContainID	Boolean	false	**true**: password must not contain the identifier of the user. **false**: password may contain the identifier of the user.

II.3 Advance grammar with the Pattern parameter (regular expressions)

The **Pattern** parameter is used to specify complex constraints on passwords by using regular expressions syntax.

This document is not a reference on regular expressions. We will give only some examples of expression that can be useful in frequent cases.

For a reference on regular expression, we are using **Java Regex** when checking the pattern.

Examples:

- All strings: `.*`
- Specify a minimum string length: `.{8,}`
- Specify a minimum and maximum string length: `.{8,20}`
- Forbid spaces and tabs: `(?=\S+$).*`
- At least one digit: `(?=.*[0-9]).*`
- At least one lower-case letter: `(?=.*[a-z]).*`
- At least one upper-case letter: `(?=.*[A-Z]).*`
- At least one special character: `(?=.*[@#$%^&+=_\-]).*`

It is possible to group these expressions in one regular expression, for instance:

```
((?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=])(?=\S+$).{8,})
```

(Note the parenthesis surrounding the combined expression)

Note: Password quality check starts with the pattern validation. If the password does not match the pattern constraints the other simple parameters ("Must...") will not be checked.