



Red Hat Directory Server 12

User management and authentication

Managing users, groups, roles, and authentication-related settings

Red Hat Directory Server 12 User management and authentication

Managing users, groups, roles, and authentication-related settings

Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Learn how to manage user access permissions, resource limits, user groups, and user roles. You can configure a password and account lockout policy, deny access for a group of users, and limit system resources depending on their bind distinguished name (bind DN).

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	3
CHAPTER 1. CHANGING THE DIRECTORY MANAGER PASSWORD	4
1.1. CHANGING THE DIRECTORY MANAGER PASSWORD USING THE COMMAND LINE	4
1.2. CHANGING THE DIRECTORY MANAGER PASSWORD USING THE WEB CONSOLE	5
CHAPTER 2. RESETTING THE DIRECTORY MANAGER PASSWORD	6
2.1. RESETTING THE DIRECTORY MANAGER PASSWORD USING THE COMMAND LINE	6
CHAPTER 3. CONFIGURING PASSWORD POLICIES	7
3.1. HOW PASSWORD POLICIES WORK	7
3.2. CONFIGURING THE GLOBAL PASSWORD POLICY USING THE COMMAND LINE	8
3.3. CONFIGURING THE GLOBAL PASSWORD POLICY USING THE WEB CONSOLE	8
3.4. LOCAL PASSWORD POLICY ENTRIES	9
3.5. CONFIGURING A LOCAL PASSWORD POLICY USING THE COMMAND LINE	11
3.6. DISABLING A LOCAL PASSWORD POLICY USING THE COMMAND LINE	12
CHAPTER 4. CONFIGURING TEMPORARY PASSWORD RULES	13
4.1. ENABLING TEMPORARY PASSWORD RULES IN THE GLOBAL PASSWORD POLICY	13
4.2. ENABLING TEMPORARY PASSWORD RULES IN A LOCAL PASSWORD POLICY	14
CHAPTER 5. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS	16
5.1. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS IN A GLOBAL POLICY	16
5.2. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS IN A LOCAL POLICY	17
5.3. ADDITIONAL RESOURCES	17
CHAPTER 6. DISABLING ANONYMOUS BINDS	18
6.1. DISABLING ANONYMOUS BINDS USING THE COMMAND LINE	18
6.2. DISABLING ANONYMOUS BINDS USING THE WEB CONSOLE	18
CHAPTER 7. MANUALLY INACTIVATING USERS AND ROLES	20
7.1. INACTIVATION AND ACTIVATION OF USERS AND ROLES USING THE COMMAND LINE	20
7.2. COMMANDS FOR DISPLAYING THE STATUS OF AN ACCOUNT OR A ROLE	20
CHAPTER 8. SYNCHRONIZING ACCOUNT LOCKOUT ATTRIBUTES ACROSS ALL SERVERS IN A REPLICATION ENVIRONMENT	23
8.1. HOW DIRECTORY SERVER HANDLES PASSWORD AND ACCOUNT LOCKOUT POLICIES IN A REPLICATION ENVIRONMENT	23
8.2. CONFIGURING DIRECTORY SERVER TO REPLICATE ACCOUNT LOCKOUT ATTRIBUTES	23
CHAPTER 9. USING REFERENTIAL INTEGRITY TO MAINTAIN RELATIONSHIPS BETWEEN ENTRIES	26
9.1. HOW THE REFERENTIAL INTEGRITY PLUG-IN WORKS	26
9.2. CONFIGURING THE REFERENTIAL INTEGRITY PLUG-IN USING THE COMMAND LINE	26
9.3. CONFIGURING THE REFERENTIAL INTEGRITY PLUG-IN USING THE WEB CONSOLE	28
CHAPTER 10. MANAGING ROLES IN DIRECTORY SERVER USING THE LDAP BROWSER	30
10.1. CREATING A ROLE IN THE LDAP BROWSER	30
10.2. DELETING A ROLE IN THE LDAP BROWSER	30
10.3. MANAGING A ROLE IN THE LDAP BROWSER	31

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
 1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
 2. Use your mouse cursor to highlight the part of text that you want to comment on.
 3. Click the **Add Feedback** pop-up that appears below the highlighted text.
 4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
 1. Go to the [Bugzilla](#) website.
 2. As the Component, use **Documentation**.
 3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
 4. Click **Submit Bug**.

CHAPTER 1. CHANGING THE DIRECTORY MANAGER PASSWORD

The Directory Manager is the privileged database administrator, comparable to the **root** user in a Linux operating system. The Directory Manager entry and the corresponding password are set during the instance installation. As an administrator, you can change the Directory Manager password to use a different one.

1.1. CHANGING THE DIRECTORY MANAGER PASSWORD USING THE COMMAND LINE

You can set a new password for the Directory Manager using the **dsconf** command line utility or manually by setting the **nsslapd-rootpw** parameter.



IMPORTANT

Set the password using an encrypted connection only. Using an unencrypted connection can expose the password to the network. If your server does not support encrypted connections, use the web console to update the Directory Manager password.

Procedure

- Set the Directory Manager password using one of the following options:
 - To encrypt the password automatically:

```
# dsconf -D "cn=Directory Manager" ldaps://server.example.com config replace
nsslapd-rootpw=password
```

Directory Server automatically encrypts the plain text value that you set in the **nsslapd-rootpw** parameter.



WARNING

Do not use curly braces **{}** in the password. Directory Server stores the password in the **{password-storage-scheme}hashed_password** format. The server interprets characters in curly braces as the password storage scheme. If the string is an invalid storage scheme or if the password is not correctly hashed, the Directory Manager cannot connect to the server.

- To encrypt the password manually:
 1. Generate a new password hash. For example:

```
# pwdhash -D /etc/dirsrv/slapd-instance_name password
{PBKDF2_SHA256}AAAgaMwPYlhEkQozTagoX6RGG5E7d6/6oOJ8TVty...
```


The password is encrypted using the password storage scheme set in the **nsslapd-rootpwstoragescheme** attribute of the Directory Server instance configuration.

2. Using a STARTTLS connection, set the **nsslapd-rootpw** attribute to the value displayed in the previous step:

```
# dsconf -D "cn=Directory Manager" Idaps://server.example.com config replace  
nsslapd-  
rootpw="{PBKDF2_SHA256}AAAgAMwPYIhEkQozTagoX6RGG5E7d6/6oOJ8TV  
ty..."
```

Additional resources

- [Changing the Directory Manager password using the web console](#)

1.2. CHANGING THE DIRECTORY MANAGER PASSWORD USING THE WEB CONSOLE

You can set a new password for the Directory Manager using the web console.

Prerequisites

- You are logged in to the instance in the web console.

Procedure

1. Open the **Server** → **Server Settings** → **Directory Manager** menu.
2. Enter the new password into the **Directory Manager Password** and **Confirm Password** fields.
3. Optional: Set a different password storage scheme.
4. Click **Save**.

CHAPTER 2. RESETTING THE DIRECTORY MANAGER PASSWORD

The Directory Manager is the privileged database administrator, comparable to the **root** user in a Linux operating system. The Directory Manager password is set during the instance installation. If you lose the password, you can reset it to regain privileged access to the directory.

2.1. RESETTING THE DIRECTORY MANAGER PASSWORD USING THE COMMAND LINE

If you have the root access to the Directory Server instance, you can reset the password of the Directory Manager.

Procedure

1. Generate a new password hash. For example:

```
# pwdhash -D /etc/dirsrv/slaped-instance_name new_password  
{PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

Because you specified the path to the Directory Server instance configuration, the **pwdhash** generator automatically uses the password storage scheme set in the **nsslapd-rootpwstoragescheme** attribute to encrypt the new password.

2. Stop the Directory Server instance:

```
# dsctl instance_name stop
```

3. Edit the **/etc/dirsrv/slaped-instance_name/dse.ldif** file and set the **nsslapd-rootpw** attribute to the value displayed in the first step:

```
nsslapd-rootpw: {PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

4. Start the Directory Server instance:

```
# dsctl instance_name start
```

CHAPTER 3. CONFIGURING PASSWORD POLICIES

A password policy minimizes the risks associated with using passwords by enforcing a certain level of security. For example, you can define a password policy to ensure that:

- Users must change their passwords according to a schedule
- Users must provide non-trivial passwords
- The password syntax must meet certain complexity requirements

3.1. HOW PASSWORD POLICIES WORK

Directory Server supports fine-grained password policies, which work in an inverted pyramid, from general to specific. A global password policy is superseded by a subtree-level password policy, which is superseded by a user-level password policy.

You can define:

- Global password policy, applied to the entire directory
- Local password policy
 - Subtree-level policy, applied to a particular subtree
 - User-level policy, applied to a particular user

Password policies are not additive: only one password policy applies to an entry. For example, when you configure a particular attribute in the global or subtree-level password policy, but not in the user-level password policy, this attribute does not apply to the user. In this case, when the user attempts to log in, only the user-level policy is active.



WARNING

When using a password administrator account or the Directory Manager (root DN) to set a password, you bypass the password policies. Do not use these accounts for regular user password management. Use them only to perform password administration tasks that require bypassing the password policies, such as adding a prehashed password, or purposefully overriding current password constraints for setting temporary passwords after a reset.

The complete password policy that applies to a user account consists of the following elements:

- **The type or level of password policy checks.** This information indicates whether the server should check for and enforce a global password policy or local password policies.
- **Password add and modify information.** The password information includes password syntax and password history details.
- **Bind information.** The bind information includes the number of grace logins permitted, password aging attributes, and tracking bind failures.

**NOTE**

After establishing a password policy, you can protect user passwords from potential threats by configuring an account lockout policy. Account lockout protects against attempts to break into the directory by repeatedly guessing a user's password.

Additional resources

- [Configuring a password-based account lockout policy](#)
- [Configuring time-based account lockout policies](#)

3.2. CONFIGURING THE GLOBAL PASSWORD POLICY USING THE COMMAND LINE

By default, global password policy settings are disabled. You can configure the global password policy using the **dsconf** command line utility.

Procedure

1. Display the current settings:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy get
Global Password Policy: cn=config
-----
passwordstorage: PBKDF2_SHA256
passwordChange: on
passwordMustChange: off
passwordHistory: off
passwordInHistory: 6
...
```

2. Adjust the password policy settings. For a full list of available settings, enter:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --help
```

For example, to enable password syntax checking and set the minimum length of passwords to **12** characters, enter:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdchecksyntax on --pwdmintokenlen 12
```

3. Enable the the account lockout feature for the password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdlockout on
```

3.3. CONFIGURING THE GLOBAL PASSWORD POLICY USING THE WEB CONSOLE

By default, global password policy settings are disabled. You can configure the global password policy using the web console.

Prerequisites

- You are logged in to the instance in the web console.

Procedure

1. Open the **Database** → **Password Policies** → **Global Policy** menu.
2. Set the global password policy settings. You can set parameters in the following categories:
 - General settings, such as the password storage scheme
 - Password expiration settings, such as the time when a password expires
 - Account lockout settings, such as after how many failed login attempts an account should be locked
 - Password syntax settings, such as the minimum password length
To display a tool tip and the corresponding attribute name in the **cn=config** entry for a parameter, hover the mouse cursor over the setting.
3. Click **Save**.

3.4. LOCAL PASSWORD POLICY ENTRIES

When you use the **dsconf localpwp addsubtree** or **dsconf localpwp adduser** commands, Directory Server automatically creates an entry to store the local password policy attributes.

For a subtree, the following entries are added:

Table 3.1. Local password policy entries for a subtree

Entry name	Description	Contents
nsPwPolicyContainer	A container entry at the subtree level	Various password policy-related entries for the subtree and all its children
nsPwPolicyEntry	The actual password policy specification entry	All the password policy attributes that are specific to the subtree
nsPwTemplateEntry	The CoS Template Entry	The pwdpolicysubentry value pointing to the nsPwPolicyEntry entry
<CoS definition entry DN>	The CoS definition entry at the subtree level	CoS definition entry

Example 3.1. The nsPwPolicyContainer entry for a subtree ou=people,dc=example,dc=com

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
```

```
cn: nsPwPolicyContainer
```

Example 3.2. The nsPwPolicyEntry entry for a subtree ou=people,dc=example,dc=com

```
dn: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
  cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: ldapsubentry
objectclass: passwordpolicy
```

Example 3.3. The nsPwTemplateEntry entry for a subtree ou=people,dc=example,dc=com

```
dn: cn="cn=nsPwTemplateEntry,ou=people,dc=example,dc=com",
  cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: costemplate
objectclass: ldapsubentry
cosPriority: 1
pwdpolicysubentry: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
  cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
```

Example 3.4. The CoS specification entry for a subtree ou=people,dc=example,dc=com

```
dn: cn=newpwdpolicy_cos,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=cn=nsPwTemplateEntry\,ou=people\,dc=example,dc=com,
  cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
cosAttribute: pwdpolicysubentry default operational
```

For a user, the following entries are added:

Table 3.2. Local password policy entries for a user

Entry name	Description	Contents
nsPwPolicyContainer	A container entry at the parent level	Various password policy-related entries for the user and all its children
nsPwPolicyEntry	The actual password policy specification entry	All the password policy attributes that are specific to the user

Example 3.5. The nsPwPolicyContainer entry for a user
uid=user_name,ou=people,dc=example,dc=com

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
cn: nsPwPolicyContainer
```

Example 3.6. The nsPwPolicyEntry entry for a user
uid=user_name,ou=people,dc=example,dc=com

```
dn: cn="cn=nsPwPolicyEntry,uid=user_name,ou=people,dc=example,dc=com",
cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: Idapsubentry
objectclass: passwordpolicy
```

3.5. CONFIGURING A LOCAL PASSWORD POLICY USING THE COMMAND LINE

In contrast to a global password policy, which defines settings for the entire directory, a local password policy is a policy for a specific user or subtree. Currently, you can only set up a local password policy using the command line.

Prerequisites

- User or subtree entries that you want to create the policy for already exist in the directory.

Procedure

1. Verify if a local password policy already exists for the subtree or user entry. For example:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp get
"ou=People,dc=example,dc=com"
Enter password for cn=Directory Manager on ldap://server.example.com:
Error: No password policy was found for this entry
```

If no local policy exists, create one:

- To create a **subtree** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp
addsubtree "ou=People,dc=example,dc=com"
```

- To create a **user** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp adduser
"uid=user_name,ou=People,dc=example,dc=com"
```

2. Set local policy attributes. For a full list of available settings, enter:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --help
```

For example, to enable password expiration and set the maximum password age to 14 days (1209600 seconds):

- On a **subtree** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --pwdexpire on --pwdmaxage 1209600 "ou=People,dc=example,dc=com"
```

- On a **user** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --pwdexpire on --pwdmaxage 1209600 "uid=user_name,ou=People,dc=example,dc=com"
```

3.6. DISABLING A LOCAL PASSWORD POLICY USING THE COMMAND LINE

When you create a new local policy, the **nsslapd-pwpolicy-local** parameter in the **cn=config** entry is automatically set to **on**. If the local password policy should not be enabled, you can disable it manually using the command line.

Procedure

- Disable all local policies or remove a particular local policy:
 - To disable all local password policies:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --pwdlocal off
```

- To remove a single existing **subtree** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp remove "ou=People,dc=example,dc=com"
```

- To remove a single existing **user** password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp remove "uid=user_name,ou=People,dc=example,dc=com"
```


CHAPTER 4. CONFIGURING TEMPORARY PASSWORD RULES

Directory Server password policies support setting temporary passwords on user accounts. If you assign a temporary password to a user, Directory Server rejects any other operation than a password change until the user changes its password.

The following are the features of temporary passwords:

- Only the **cn=Directory Manager** account can assign temporary passwords.
- Directory Server allows authentication attempts only for a fixed number of times to avoid that an attacker probes the password.
- Directory Server allows authentication attempts after a specified delay to configure that the temporary passwords are not usable directly after you set them.
- Directory Server allows authentication attempts only for a specified time so that the temporary password expires if a user does not use or reset it.
- If the authentication was successful, Directory Server requires that the user resets the password before the server performs any other operation.

By default, temporary password rules are disabled. You can configure them in global or local password policies.

4.1. ENABLING TEMPORARY PASSWORD RULES IN THE GLOBAL PASSWORD POLICY

To enable the temporary password feature for the whole Directory Server instance:

1. Enable that users must change their password if an administrator resets it.
2. Configure the feature in the global password policy.

If an administrator updates the **userPassword** attribute of a user and sets the **passwordMustChange** attribute to **on**, Directory Server applies the temporary password rules.

Procedure

1. Configure that a user must change its password after an administrator resets it:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwmustchange on
```

2. Configure the temporary password rules settings in a global password policy:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60
```

In this example:

- The **--pwptprmaxuse** option sets the maximum number of attempts a user can use the temporary password to **5**.

- The **--pwptprdelayexpireat** option sets the time before the temporary password expires to **3600** seconds (1 hour).
- The **--pwptprdelayvalidfrom** option configures that the time set in **--pwptprdelayexpireat** starts **60** seconds after an administrator reset the password of a user.

Verification

- Display the attributes that store the temporary password rules:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy get | grep -i TPR
passwordTPRMaxUse: 5
passwordTPRDelayExpireAt: 3600
passwordTPRDelayValidFrom: 60
```

4.2. ENABLING TEMPORARY PASSWORD RULES IN A LOCAL PASSWORD POLICY

To enable the temporary password feature for a specific user or sub-tree, enable that users must change their password if an administrator resets it, and configure the feature in a local password policy.

If an administrator updates the **userPassword** attribute of a user and sets the **passwordMustChange** attribute to **on**, Directory Server applies the temporary password rules if the user:

- Has the local password policy enabled
- Is stored in a sub-tree that has the local password policy enabled

Procedure

1. Configure that a user must change its password after an administrator resets it:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --pwmustchange on
```

2. Configure the temporary password rules settings:

- For an existing sub-tree:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp addsubtree --pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60 ou=People,dc=example,dc=com
```

- For an existing user:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp adduser -pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60 uid=example,ou=People,dc=example,dc=com
```

In these examples:

- The **--pwptprmaxuse** option sets the maximum number of attempts a user can use the temporary password to **5**.

- The **--pwptprdelayexpireat** option sets the time before the temporary password expires to **3600** seconds (1 hour).
- The **--pwptprdelayvalidfrom** option configures that the time set in **--pwptprdelayexpireat** starts **60** seconds after an administrator reset the password of a user.

Verification

- Display the local password policy of the distinguished name (DN):

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp get <DN> |  
grep -i TPR  
passwordTPRMaxUse: 5  
passwordTPRDelayExpireAt: 3600  
passwordTPRDelayValidFrom: 60
```

CHAPTER 5. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS

The Directory Manager can assign the *password administrator* role to a user or a group of users. Because password administrators need access control instructions (ACIs) with the appropriate permissions, Red Hat recommends that you configure a group to allow a single ACI set to manage all password administrators.

Using the password administrator role is beneficial in the following scenarios:

- setting up an attribute that forces the user to change their password at the time of the next login
- changing a user's password to a different storage scheme defined in the password policy



IMPORTANT

A password administrator can perform any user password operations. When using a password administrator account or the Directory Manager (root DN) to set a password, password policies are bypassed and not verified. Do not use these accounts for regular user password management. Red Hat recommends performing ordinary password updates under an existing role in the database with permissions to update only the **userPassword** attribute.



NOTE

You can add a new **passwordAdminSkipInfoUpdate: on/off** setting under the **cn=config** entry to provide a fine grained control over password updates performed by password administrators. When you enable this setting, passwords updates do not update certain attributes, for example, **passwordHistory**, **passwordExpirationTime**, **passwordRetryCount**, **pwdReset**, and **passwordExpWarned**.

5.1. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS IN A GLOBAL POLICY

In a global policy, you can assign the password administrator role to a user or a group of users. Red Hat recommends that you configure a group to allow a single access control instruction (ACI) set to manage all password administrators.

Prerequisites

- You have created a group named **password_admins** that includes all of the users to whom you want to assign the password administrator role.

Procedure

1. Create the ACI that defines the permissions for a password administrator role:

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x dn:
ou=people,dc=example,dc=com changetype: modify add: aci aci:
(targetattr="userPassword || nsAccountLock || userCertificate || nsSshPublicKey")
```

```
(targetfilter="(objectClass=nsAccount"))(version 3.0; aci "Enable user password
reset"; allow (write, read)
(groupdn="ldap:///cn=password_admins,ou=groups,dc=example,dc=com");)
```

2. Assign the password administrator role to the group:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdadmin "cn=password_admins,ou=groups,dc=example,dc=com"
```

5.2. ASSIGNING PASSWORD ADMINISTRATOR PERMISSIONS IN A LOCAL POLICY

In a local policy, you can assign the password administrator role to a user or a group of users. Red Hat recommends that you configure a group to allow a single access control instruction (ACI) set to manage all password administrators.

Prerequisites

- You have created a group named **password_admins** that includes all of the users to whom you want to assign the password administrator role.

Procedure

1. Create the ACI that defines the permissions for a password administrator role:

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x dn:
ou=people,dc=example,dc=com changetype: modify add: aci aci:
(targetattr="userPassword || nsAccountLock || userCertificate || nsSshPublicKey")
(targetfilter="(objectClass=nsAccount"))(version 3.0; aci "Enable user password
reset"; allow (write, read)
(groupdn="ldap:///cn=password_admins,ou=groups,dc=example,dc=com");)
```

2. Assign the password administrator role to the group:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set
ou=people,dc=example,dc=com --pwdadmin
"cn=password_admins,ou=groups,dc=example,dc=com"
```

5.3. ADDITIONAL RESOURCES

- [Backup up Directory Server \(change it\)](#)

CHAPTER 6. DISABLING ANONYMOUS BINDS

If a user attempts to connect to Directory Server without supplying any credentials, this operation is called **anonymous bind**. Anonymous binds simplify searches and read operations, such as finding a phone number in the directory by not requiring users to authenticate first. However, anonymous binds can also be a security risk, because users without an account are able to access the data.



WARNING

By default, anonymous binds are enabled in Directory Server for search and read operations. This allows unauthorized access to user entries as well as configuration entries, such as the root directory server entry (DSE).

6.1. DISABLING ANONYMOUS BINDS USING THE COMMAND LINE

To increase the security, you can disable anonymous binds.

Procedure

- Set the **nsslapd-allow-anonymous-access** configuration parameter to **off**:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-allow-anonymous-access=off
```

Verification

- Run a search without specifying a user account:

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
additional info: Anonymous access is not allowed
```

6.2. DISABLING ANONYMOUS BINDS USING THE WEB CONSOLE

To increase the security, you can disable anonymous binds.

Prerequisites

- You are logged in to the instance in the web console.

Procedure

1. Navigate to **Server** → **Server Settings** → **Advanced Settings**.
2. Set the **Allow Anonymous Access** parameter to **off**.
3. Click **Save**.

Verification

- Run a search without specifying a user account:

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
  additional info: Anonymous access is not allowed
```

CHAPTER 7. MANUALLY INACTIVATING USERS AND ROLES

In Directory Server, you can temporarily inactivate a single user account or a set of accounts. Once an account is inactivated, a user cannot bind to the directory. The authentication operation fails.

7.1. INACTIVATION AND ACTIVATION OF USERS AND ROLES USING THE COMMAND LINE

You can manually inactivate users and roles using the command line or the operational attribute.

Roles behave as both a static and a dynamic group. With a group, entries are added to a group entry as members. With a role, the role attribute is added to an entry and then that attribute is used to identify members in the role entry automatically.

Users and roles are inactivated executing the same procedures. However, when a role is inactivated, the members of the role are inactivated, not the role entry itself.

To inactivate users and roles, execute the following commands in the command line:

- For inactivation of a user account:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account lock "uid=user_name,ou=People,dc=example,dc=com"
```

- For inactivation of a role:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role lock "cn=Marketing,ou=People,dc=example,dc=com"
```

To activate users and roles, execute the following commands in the command line:

- For activation of a user account:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account unlock
"uid=user_name,ou=People,dc=example,dc=com"
```

- For activation of a role:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role unlock "cn=Marketing,ou=People,dc=example,dc=com"
```

Optionally, instead of using the commands, you can add the operational attribute **nsAccountLock** to the entry. When an entry contains the **nsAccountLock** attribute with a value of **true**, the server rejects the bind.

7.2. COMMANDS FOR DISPLAYING THE STATUS OF AN ACCOUNT OR A ROLE

You can display the status of an account or a role in Directory Server using the corresponding commands in the command line.

Commands for displaying the status

- Display the status of an account:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com"
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210813085535Z (2021-08-13 08:55:35)
Entry Modification Date: 20210813085535Z (2021-08-13 08:55:35)
Entry State: activated
```

Optional: The **-V** option displays additional details.

Example 7.1. Detailed output for an active account

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com" -V
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Modification Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Last Login Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Time Until Inactive: 2 seconds (2021-08-24 16:07:45)
Entry State: activated
```

Example 7.2. Detailed output for an inactive account

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com" -V
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Modification Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Last Login Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Time Since Inactive: 3 seconds (2021-08-24 16:07:45)
Entry State: inactivity limit exceeded
```

- Display the status of a role:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role entry-status
"cn=Marketing,ou=People,dc=example,dc=com"
Entry DN: cn=Marketing,ou=people,dc=example,dc=com
Entry State: activated
```

- Display the status of a sub-tree:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account subtree-status "ou=People,dc=example,dc=com" -f
"(uid=*)" -V -o "2021-08-25T14:30:30"
```

To filter the results of the search in a sub-tree, you can use:

- The **-f** option to set the search filter
- The **-s** option to set the search scope
- The **-i** option to return only inactive accounts
- The **-o** option to return only accounts which will be inactive before the specified date **YYYY-MM-DDTHH:MM:SS**

CHAPTER 8. SYNCHRONIZING ACCOUNT LOCKOUT ATTRIBUTES ACROSS ALL SERVERS IN A REPLICATION ENVIRONMENT

Directory Server stores account lockout attributes locally. In an environment with multiple servers, configure replication for these attributes to prevent attackers from attempting to log in to one server until the account lockout count is reached and then continue on other servers.

8.1. HOW DIRECTORY SERVER HANDLES PASSWORD AND ACCOUNT LOCKOUT POLICIES IN A REPLICATION ENVIRONMENT

Directory Server enforces password and account lockout policies as follows:

- Password policies are enforced on the data supplier
- Account lockout policies are enforced on all servers in a replication topology

Directory Server replicates the following password policy attributes:

- **passwordMinAge**
- **passwordMaxAge**
- **passwordExp**
- **passwordWarning**

However, by default, Directory Server does not replicate the general account lockout attributes:

- **passwordRetryCount**
- **retryCountResetTime**
- **accountUnlockTime**

To prevent attackers from attempting to log in to one server until the account lockout count is reached and then continue on other servers, replicate these account lockout attributes.

Additional resources

- [Configuring Directory Server to replicate account lockout attributes](#)

8.2. CONFIGURING DIRECTORY SERVER TO REPLICATE ACCOUNT LOCKOUT ATTRIBUTES

If you use an account lockout policy or password policy that updates the **passwordRetryCount**, **retryCountResetTime**, or **accountUnlockTime** attributes, configure Directory Server to replicate these attributes so that their values are the same across all servers.

Perform this procedure on all suppliers in the replication topology.

Prerequisites

- You configured an account lockout policy or a password policy that updates one or more of the mentioned attributes.
- You use Directory Server in a replication environment.

Procedure

1. Enable replication of password policy attributes:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwwisglobal="on"
```

2. If you use fractional replication, display the list of attributes that are excluded from replication:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt get --suffix
"dc=example,dc=com" example-agreement | grep "nsDS5ReplicatedAttributeList"
```

Using the default settings, no output is shown, and Directory Server replicates the account lockout attributes. However, if the command returns a list of excluded attributes, such as in the following example, verify the attribute list:

```
nsDS5ReplicatedAttributeList: (objectclass=*) $ EXCLUDE accountUnlockTime
passwordRetryCount retryCountResetTime example1 example2
```

In this example, the **accountUnlockTime**, **passwordRetryCount**, and **retryCountResetTime** lockout policy attributes are excluded from replication, along with two other attributes.

3. If the output of the previous command lists any of the account lockout attributes, update the fractional replication settings to only include attributes other than the lockout policy attributes:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt set --suffix
"dc=example,dc=com" --frac-list "example1 example2" example-agreement
```

Verification

1. Attempt to perform a search as a user using an invalid password:

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w "invalid-password" -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)
```

2. Display the **passwordRetryCount** attribute of the user:

```
# ldapsearch -H ldap://server.example.com -D "cn=Directory Manager" -W -b
"uid=example,ou=People,dc=example,dc=com" -x passwordRetryCount
...
dn: uid=example,ou=People,dc=example,dc=com
passwordRetryCount: 1
```

3. Run the previous command on a different server in the replication topology. If the value of the **passwordRetryCount** attribute is the same, Directory Server replicated the attribute.

Additional resources

Additional resources

- [Configuring a password-based account lockout policy](#)

CHAPTER 9. USING REFERENTIAL INTEGRITY TO MAINTAIN RELATIONSHIPS BETWEEN ENTRIES

Referential Integrity is a database mechanism that ensures that Directory Server maintains relationship between related entries. You can use this feature to ensure that an update to one entry in the directory is correctly reflected in other entries that reference the updated entry.

For example, if you remove a user from the directory and the Referential Integrity plug-in is enabled, the server also removes the user from any group in which the user is a member. If the plug-in is not enabled, the user remains a member of the group until an administrator manually removes it.

Referential Integrity is an important feature if you integrate Directory Server with other products that rely on Directory Server for user and group management.

9.1. HOW THE REFERENTIAL INTEGRITY PLUG-IN WORKS

When you enable the Referential Integrity plug-in, it performs integrity updates on the **member**, **uniqueMember**, **owner**, and **seeAlso** attributes, by default, immediately after an operation.

For example, if an administrator deletes, updates, renames, or moves a group or user within the directory, Directory Server logs the operation in the Referential Integrity log file. Directory Server then uses the distinguished name (DN) from this log file and searches entries matching the attribute specified in the plug-in's configuration, and then updates the matching entries. For example, after deleting the **cn=demo,dc=example,dc=com** entry the plug-in searches for entries with the **member** attribute set to **cn=demo,dc=example,dc=com** and removes these **member** attributes. Afterwards, the plug-in does the same for the **uniqueMember**, **owner**, and **seeAlso** attributes.

By default, Directory Server does searches and updates in the same transaction as the original operation. Because search and update operations can take a lot of time, it is possible to delay them after the completion of the original operation. You can use the **--update-delay** option of the **dsconf plugin referential-integrity set** command to separate the original operations from integrity updates.

To avoid poor performance of modify and delete operations, index the attributes you specify in the Referential Integrity plug-in configuration.

Additional resources

- [Managing indexes](#)

9.2. CONFIGURING THE REFERENTIAL INTEGRITY PLUG-IN USING THE COMMAND LINE

You can use the command line to configure the Referential Integrity plug-in.

Perform this procedure on every supplier in a replication topology.

Procedure

1. Enable the Referential Integrity plug-in:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity enable
```

2. Set the subtree in which the plug-in searches for delete or rename operations of user entries:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --entry-scope "ou=People,dc=example,dc=com"
```

3. Optional: Exclude a subtree under the entry scope:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --exclude-entry-scope "ou=Special Users,ou=People,dc=example,dc=com"
```

This command configures the plug-in to ignore delete or rename operations performed in the **ou=Special Users,ou=People,dc=example,dc=com** subtree.

4. Configure the subtree in which the plug-in updates group entries:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --container-scope "ou=Groups,dc=example,dc=com"
```

5. By default, the plug-in performs integrity updates on the **member**, **uniqueMember**, **owner**, and **seeAlso** attributes. To specify other attributes, enter:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --membership-attr attribute_1 attribute_2
```

Note that this command overrides the list of attributes in the plug-in's configuration. If you want to add an attribute, pass the current list of attributes and the additional one to the **--membership-attr** option.

6. Optional: By default, Directory Server performs referential integrity checks immediately. If you want to set a delay, enter:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --update-delay=5
```

This command delays the referential integrity checks by **5** seconds. Note that, if you enabled the Referential Integrity on multiple suppliers, setting a delay can cause replication loops and directory inconsistencies. To avoid such problems, enable the plug-in only on one supplier in the topology.

7. Restart the instance:

```
# dsctl instance_name restart
```

Verification

1. Display the Referential Integrity plug-in configuration:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity show
...
nsslapd-plugincontainerscope: ou=Groups,dc=example,dc=com
nsslapd-pluginentryscope: ou=People,dc=example,dc=com
...
```

```
referint-membership-attr: member
referint-membership-attr: uniquemember
referint-membership-attr: owner
referint-membership-attr: seeAlso
referint-update-delay: 0
...
```

- List the members of a group by displaying the **member** attributes of the groups:

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=Groups,dc=example,dc=com" member
...
member: uid=demoUser,ou=People,dc=example,dc=com
```

- Delete the **uid=demoUser,ou=People,dc=example,dc=com** user:

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" user delete "uid=demoUser,ou=People,dc=example,dc=com"
```

- Display the members of the group again:

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=People,dc=example,dc=com" member
```

If **uid=demoUser,ou=People,dc=example,dc=com** is no longer listed as a member of the group, the Referential Integrity plug-in works.

9.3. CONFIGURING THE REFERENTIAL INTEGRITY PLUG-IN USING THE WEB CONSOLE

You can use the Directory Server web console to configure the Referential Integrity plug-in.

Perform this procedure on every supplier in a replication topology.

Prerequisites

- You are logged in to the instance in the web console.

Procedure

- Navigate to **Plugins** → **Referential Integrity**.
- Enable the plug-in.
- Click **Actions** → **Restart Instance**.
- Navigate again to **Plugins** → **Referential Integrity**.
- By default, the plug-in performs integrity updates on the **member**, **uniqueMember**, **owner**, and **seeAlso** attributes. To specify other attributes, update the list in the **Membership Attribute** field.
- Set the **Entry Scope** field to the DN of the subtree in which the plug-in should search for delete or rename operations of user entries.

7. Optional: To exclude a subtree under the entry scope, enter the DN of the subtree in the **Exclude Entry Scope** field.
8. Set the **Container Scope** field to the DN of the subtree in which the plug-in should update group entries.
9. Optional: Update the path to the Referential Integrity log file. Directory Server uses this file to track changes in the directory. Note that the **dirsrv** user must have write permissions to this location.
10. Optional: By default, Directory Server performs referential integrity checks immediately. If you want to set a delay, set it in the **Update Delay** field.
Note that, if you enabled the Referential Integrity on multiple suppliers, setting a delay can cause replication loops and directory inconsistencies. To avoid such problems, enable the plug-in only on one supplier in the topology.
11. Click **Save Config**.

Verification

1. List the members of a group by displaying the **member** attributes of the groups:

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=Groups,dc=example,dc=com" member
...
member: uid=demoUser,ou=People,dc=example,dc=com
```

2. Delete the **uid=demoUser,ou=People,dc=example,dc=com** user:

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" user delete "uid=demoUser,ou=People,dc=example,dc=com"
```

3. Display the members of the group again:

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=People,dc=example,dc=com" member
```

If **uid=demoUser,ou=People,dc=example,dc=com** is no longer listed as a member of the group, the Referential Integrity plug-in works.

CHAPTER 10. MANAGING ROLES IN DIRECTORY SERVER USING THE LDAP BROWSER

A role is a grouping mechanism that unifies static and dynamic groups.

10.1. CREATING A ROLE IN THE LDAP BROWSER

You can create a role for a Red Hat Directory Server entry by using the **LDAP Browser** wizard in the web console.

Prerequisites

- Access to the web console.
- A parent entry exists in the Red Hat Directory Server.

Procedure

1. Log in to the web console and click **Red Hat Directory Server**.
2. After the web console loads the **Red Hat Directory Server** interface, open the **LDAP Browser**.
3. Select an LDAP entry and open the **Options** menu.
4. From the drop-down menu select **New** and click **Create a new role**.
5. Follow the steps in the wizard and click the **Next** button after you complete each step.
6. To create the role, review the role settings in the **Create Role** step and click the **Create** button. You can click the **Back** button to modify the role settings or click the **Cancel** button to cancel the role creation.
7. To close the wizard window, click the **Finish** button.

Verification

- Expand the LDAP entry and verify the new role appears among the entry parameters.

10.2. DELETING A ROLE IN THE LDAP BROWSER

You can delete the role from the Red Hat Directory Server entry by using the **LDAP Browser** in the web console.

Prerequisites

- Access to the web console.
- A parent entry exists in the Red Hat Directory Server.

Procedure

1. Log in to the web console and click **Red Hat Directory Server**.
2. After the web console loads the **Red Hat Directory Server** interface, click **LDAP browser**.

3. Expand the LDAP entry select the role which you want to delete.
4. Open the **Options** menu and select **Delete**.
5. Verify the data about the role you want to delete and click the **Next** button until you reach the **Deletion** step.
6. Toggle the switch to the **Yes, I'm sure** position and click the **Delete** button.
7. To close the wizard window, click the **Finish** button.

Verification

- Expand the LDAP entry and verify the role is no longer a part of the entry parameters.

10.3. MANAGING A ROLE IN THE LDAP BROWSER

You can modify the role parameters for a Red Hat Directory Server entry by using the **LDAP Browser** in the web console.

Prerequisites

- Access to the web console.
- A parent entry exists in the Red Hat Directory Server.

Procedure

1. Log in to the web console and click **Red Hat Directory Server**.
2. After the web console loads the **Red Hat Directory Server** interface, click **LDAP Browser**.
3. Expand the LDAP entry and select the role you are modifying.
4. Click the **Options** menu and select **Edit** to modify the parameters of the role or **Rename** to rename the role.
5. In the wizard window modify the necessary parameters and click **Next** after each step until you observe the **LDIF Statements** step.
6. Check the updated parameters and click **Modify Entry** or **Change Entry Name**.
7. To close the wizard window, click the **Finish** button.

Verification

- Expand the LDAP entry and verify the updated parameters are listed for the role.