Red Hat Directory Server Red Hat Directory Server 9 Updates Available in Red Hat Enterprise Linux 6.4

Enhancements to the 389-ds Package
Edition 9.0.1

Ella Deon Lackey

Red Hat Directory Server Red Hat Directory Server 9 Updates Available in Red Hat Enterprise Linux 6.4

Enhancements to the 389-ds Package
Edition 9.0.1

Ella Deon Lackey
dlackey@redhat.com

Legal Notice

Abstract

The 389-ds-base package, and several dependencies, have been updated in Red Hat Enterprise Linux 6.4. These new packages provide additional functionality that was not available in the initial release of Red Hat Directory Server 9.0 on Red Hat Enterprise Linux 6.2 or in subsequent updates. Directory Server 9.0 instances still running on Red Hat Enterprise Linux 6.2 or 6.3 will not have these new features. The enhancements to the 389-ds-base package does not affect the Directory Server Console or the Administration Server. This documentation is no longer maintained. For details, see .

Table of Contents

Red Hat Enterprise Linux 6.2, 6.3, and 6.4 all include the core *389-ds-base* package as part of the distribution. This package provides the central Directory Server functionality and comprises the Directory Server instance and associated tools. Red Hat Directory Server 9.0.1 includes extended usability on top of the *389-ds-base* through the Directory Server Console and Red Hat Administration Server and other supporting tools.

Red Hat Enterprise Linux 6.4 includes enhancements to the *389-ds-base* package which introduce some new and enhanced features. When the underlying operating system is upgraded to Red Hat Enterprise Linux 6.4, any installed Directory Server instance will receive this new functionality because of the upgraded *389-ds-base* package.

This document covers the enhancements included in the Red Hat Enterprise Linux 6.4 release. Any Directory Server 9.0.1 instances running on Red Hat Enterprise Linux 6.0, 6.1, or 6.2 will not have this enhanced functionality.

1. Deprecated Documentation

Important

Note that as of June 10, 2017, the support for Red Hat Directory Server 9 has ended. For details, see "*Red Hat Directory Server Life Cycle policy*". Red Hat recommends users of Directory Server 9 to update to the latest version.

Due to the end of the maintenance phase of this product, this documentation is no longer updated. Use it only as a reference!

2. Release Information for Red Hat Directory Server and 389-ds-base

- Red Hat Directory Server Updates for Red Hat Enterprise Linux 6.4
- Red Hat Directory Server Updates for Red Hat Enterprise Linux 6.3
- Red Hat Enterprise Linux 6.3 Technical Notes
- Red Hat Directory Server 9.0 Release Notes

3. Enhancements Available in Red Hat Enterprise Linux 6.4

3.1. Auto Membership Updates on Existing Entries

The Auto Membership Plug-in was introduced in Red Hat Directory Server 9.0 as a way to add a user to a group automatically when that user was created. By design, this only applied to new users, as they were created.

In Directory Server 9, new tasks have been added which allow the Auto Membership Plug-in to run against existing users:

- *rebuild membership*, which re-runs the Auto Membership Plug-in on existing entries to update the group membership; this is essentially a fix-up task
- *automember export updates*, which does a test-run of what the membership changes would be and writes them to a specified LDIF file

- *map updates*, which inputs the entries from an LDIF file, performs a test-run, and then writes what the results of the fix-up task would be to a given LDIF file

This can ease group administration as desired membership policies are created or modified.

3.2. PAM Pass-Through Authentication Rules per Directory Suffix

PAM pass-through authentication allows a user authenticating to the Directory Server to be authenticated against the system credentials defined in a PAM module. (The authentication request is *passed-through* to the other identity provider.)

Previously, PAM pass-through authentication was enabled for and defined for the entire directory. Starting in Red Hat Directory Server 9, PAM pass-through authentication can be defined for entries within a suffix of the directory, rather than the entire directory.

The entries to which PAM pass-through applies are identified with the ***pamFilter*** plug-in attribute in the PAM Pass-Through Authentication Plug-in configuration entry. This allos a search filter which can target a specific user, specific attribute and value, or a suffix in the tree.

Because of the new flexibility in where to apply PAM pass-through authentication rules, the PAM Pass-Through Authentication Plug-in has been enhanced to allow multiple configuration entries. This allows there to be multiple directory filters, mapping methods, and other settings, depending on specific needs within the directory.

3.3. Setting an ACI for the Directory Manager

Access control instructions are set at different points in the directory tree, at the root suffix, subsuffixes, or even user (leaf) levels. ACIs can also be set on configuration suffixes, such as **cn=config**. However, with that structure for ACI targets, it was not possible to set any ACI on the Directory Manager user because that is a special user which exists outside the directory tree.

Beginning with this update, there is a new plug-in, RootDN Access Control Plug-in, which defines some access control rules for the root user (Directory Manager) account. This allows restrictions based on time of day and host or IP address of the source machine, for enhanced security.

3.4. Option to Disable Replication Agreements

In previous versions of Directory Server there was no explicit way to disable a replication agreement. The only methods to suspend replication were to change the schedule or to delete the agreement entirely.

The ***nsds5ReplicaEnabled*** attribute is introduced in this update, which works as a switch to enable or disable a replication agreement. This allows a master server to be removed from the topology for maintenance or updates without having to completely dismantle the replication configuration for that server.

3.5. New CLEANRUV Clean-up Task

When a server is removed from the replication topology, the metadata for that server and its update operations (the RUV) still remain on the other servers. Retaining that server and RUV information can create unstable behavior with other master servers in certain situations.

Two new tasks have been introduced for cleaning out lingering RUVs:

- CLEANRUV, which removes the RUVs for the specified replica on a single master
- CLEANALLRUV, which removes the RUVs for the specified replica on a master **and then replicates that operation to all other masters and hubs in the replication topology**

These tasks can be initiated by creating a task entry in **cn=tasks**. Alternatively, the task to remove a dead replica from the RUV of another replica can be initiated by adding an attribute for the task to the second replica entry:

```
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
...
nsds5task: CLEANALLRUVreplicaID
```

3.6. Account Usability Control for LDAP Searches

This update to Directory Server introduced a new **ldapsearch** control which allows credential-less bind attempts to return authentication information about an account.

Most of the time, for the Directory Server to return authentication information about a user account, a client actually binds (or attempts to bind) as that user. And a bind attempt requires some sort of user credentials, usually a password or a certificate. While the Directory Server allows unauthenticated binds and anonymous binds, neither of those binds returns any user account information.

There are some situations where a client requires information about a user account — specifically whether an account should be allowed to authenticate — in order to perform some other operation, but the client either does not have or does use any credentials for the user account in Directory Server. Essentially, the client needs to perform a credential-less yet authenticated bind operation to retrieve the user account information (including password expiration information, if the account has a password).

This can be done through an **ldapsearch** by passing the *Account Usability Extension Control*. This control acts as if it performs an authenticated bind operation for a given user and returns the account status for that user, but without actually binding to the server. This allows a client to determine whether that account can be used to log in and then to pass that account information to another application, like PAM.

Important

The OpenLDAP tools used by Directory Server do not support the Account Usability Extension Control. Other LDAP utilities, like OpenDS, or other clients which do support the control can be used.

3.7. Operational Attribute for the Last Password Change Time

Previously, any change to an entry updated the ***modifyTimeStamp*** operational attribute.

Starting with this update, a new operational attribute, ***pwdUpdateTime***, is used specifically for the last modify time of passwords, separate from the overall entry modify time. This gives password and lockout policies another attribute to evaluate for account inactivity or password expiration.

Tracking the last password change time is enabled in the ***passwordTrackUpdateTime*** configuration attribute.

3.8. Enhanced Simple Paged Results

Simple paged results break search results into smaller chunks, with a certain number of results per page, to make it easier to read and browse search results.

This update allows simple paged results (an **ldapsearch** control) to be combined with server-side sorting on results (another **ldapsearch** control).

3.9. Tracking Changes by Bind DN

In previous versions of Directory Server, whenever an entry was created or modified, there was an operational attribute added to the entry with the modifying entry's name. However, the ***creatorsName*** and ***modifiersName*** attributes only showed whatever entry directly edited the entry. If an entry was edited through a plug-in — such as the MemberOf Plug-in updating a user entry when a group entry is edited — ther the ***modifiersName*** attribute showed the plug-in name — but not the identity of whatever user triggered the plug-in operation.

The new ***internalModifiersName*** operational attribute shows the name of whatever Directory Server plug-in performed an operation, while the ***modifiersName*** attribute now reflects whatever *bound user* initiated the operation.

```
dn: uid=bjensen,ou=people,dc=example,dc=com
...
modifiersname: uid=jsmith,ou=people,dc=example,dc=com
internalModifiersname: cn=memberof plugin,cn=plugins,cn=config
```

Tracking the bind DN is enabled by setting the ***nsslapd-plugin-binddn-tracking*** configuration attribute.

3.10. Synchronization of Posix Attributes

On Linux systems, system users and groups are identified as Posix entries, and LDAP Posix attributes contain that required information. When Windows users are synced over, they have ***ntUser*** and ***ntGroup*** attributes automatically added which identify them as Windows accounts. However, by default, no Posix attributes are synced over (even if they exist on the Active Directory entry) and no Posix attributes are added on the Directory Server side.

The new Posix Winsync API Plug-in synchronizes Posix attributes set on Active Directory over to the corresponding Directory Server entry.

Note

Syncing Posix attributes is uni-directional. **It syncs Posix attributes from Active Directory entries to Directory Server entries.** An Posix attributes added or modified on a Directory Server entry are not synced over to the corresponding Active Directory entry.

3.11. Extended Support for IPv6 for Directory Operations

Initial support for IPv6 addresses was added in Red Hat Directory Server 9.0, but not all directory operations could be performed over IPv6. This update extends IPv6 support to the remaining directory tasks, including replication, installation, chaining databases, and access control instructions.

3.12. New Disk Monitoring

If the disk space on the system where the Directory Server is running reaches a critical point, the **slapd** process can crash and, potentially, corrupt the directory database and lose information.

These updates to Directory Server introduce the ability to monitor disk space in the partition or mount point where the **slapd** process is running. If the disk space reaches an administrator-defined threshold, then the **slapd** process shuts down gracefully — preserving the database and directory data.

A new set of configuration options, ***nsslapd-disk-monitoring****, set whether disk monitoring is enabled for the **slapd** process and the disk space thresholds, grace periods for disk space warnings, and whether to lower logging levels and disable logging before shutting down the server.

3.13. Option to Disable Legacy-Style Password Lockouts

There are different ways of interpreting when the maximum password failure (***passwordMaxFailure***) has been reached. It depends on how the server counts the last failed attempt in the overall failure count.

The traditional behavior for LDAP clients is to assume that the failure occurs after the limit has been reached. So, if the failure limit is set to three, then the lockout happens at the fourth failed attempt. This also means that if the fourth attempt is successful, then the user can authenticate successfully, even though the user technically hit the failure limit. This is basically *n+1* on the count.

LDAP clients increasingly expect the maximum failure limit to look at the last failed attempt in the count as the final attempt. So, if the failure limit is set to three, then at the third failure, the account is locked. A fourth attempt, even with the correct credentials, fails. This is *n* on the count.

A new attribute, ***passwordLegacyPolicy***, is available in Red Hat Directory Server to disable the legacy password policy behavior and allow newer LDAP clients to interact properly with password policies in Directory Server.

3.14. Support for PLAIN Mechanism with SASL Authentication

Support has been added for the PLAIN mechanism with SASL authentication. While not recommended for most situations, SASL/PLAIN can be useful for anonymous connections or for times when a UID, rather than a DN, is used for authentication.

3.15. Changed DNA Plug-in Configuration

Previously, the Distributed Number Assignment (DNA) Plug-in was configured in a plug-in entry with the **extensibleObject** object class. The schema has been enhanced so that there is a new, plug-in specific object class called **dnaPluginConfig**.

3.16. Replication Option to Reject Modfications to Specified Attributes

Changes to some attributes may not represent real changes to an entry — such as an automatic change to the modify timestamp attribute from a plug-in, when no other attributes were modified.

To reduce replication load and to improve the performance of some applications, it is beneficial to ignore changes to those attributes when replicating entries. A list of attributes can be configured which can be ignored when the server evaluates entries to enter into the changelog and trigger replication. This list is defined in the ***nsds5ReplicaStripAttrs*** attribute on the replication agreement entry.

3.17. Option to Set TLSv1 for Secure Connections

Previously, both SSL encryption and TLS encryption were set together and defined with the same (SSL-specific) attributes. A new attribute, ***nsTLS1***, has been introduced which enables TLS connections, independently of SSL connections. This allows the stronger TLS protocol to be used and SSL to be disabled.