FORGEROCK

Release Notes
/ Directory Services 5.5

Latest update: 5.5.3

Mark Craig

Copyright © 2011-2018 ForgeRock AS.

Abstract

Notes covering ForgeRock® Directory Services features, fixes, and known issues.

Table of Contents

About Directory Services Software

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

The ForgeRock Common REST API works across the platform to provide common ways to access web resources and collections of resources.

Directory Services software provides an LDAPv3-compliant directory service, developed for the Java platform, delivering a high-performance, highly available, and secure store for the identities managed by your organization. *Read these notes before you install or upgrade Directory Services software.*

The easy installation process, combined with the power of the Java platform, makes this the simplest and fastest directory service to deploy and manage. Directory Services software comes with plenty of tools. Directory Services software also offers REST access to directory data over HTTP.

Directory Services software is free to download, evaluate, and use for developing your applications and solutions. ForgeRock offers training and support subscriptions to help you get the most out of your deployment.

These release notes cover the following topics:

• Hardware and software prerequisites for installing and upgrading Directory Services software

• Compatibility with previous releases

• Potential upcoming deprecation and removals that affect scripts and applications

• Issues fixed since the previous release

• Known issues open at the time of release

• Documentation updates

See the Installation Guide after you read these *Release Notes*. The *Installation Guide* also covers upgrade for Directory Services software.

Chapter 1

What's New

This chapter covers new capabilities in Directory Services 5.5, and subsequent maintenance releases.

1.1. Maintenance Releases

ForgeRock maintenance releases contain a collection of fixes that have been grouped together and released as part of our commitment to support our customers. For general information on ForgeRock's maintenance and patch releases, see Maintenance and Patch Availability Policy.

DS 5.5.3

- DS 5.5.3 is the latest release targeted for DS 5.5.0, DS 5.5.1, or DS 5.5.2 deployments and can be downloaded from the *ForgeRock Backstage* website.

  The release can be deployed as an initial deployment or updated from an existing 5.5.x deployment. DS 5.5.0, DS 5.5.1, or DS 5.5.2 is available for download at the *ForgeRock Backstage* website.

1.2. What's New in 5.5

DS 5.5.3

- There are no new features introduced in DS 5.5.3, only bug fixes.

DS 5.5.2

- There are no new features introduced in DS 5.5.2, only bug fixes.

DS 5.5.1

- There are no new features introduced in DS 5.5.1, only bug fixes.

DS 5.5

This release of Directory Services software includes the following new features:

- Backend Database Storage

JE backend databases are upgraded to Berkeley DB Java Edition 7.4.5 in this release.

• Directory Proxy

Directory proxy servers now automatically retry operations when they detect a temporary failure on the remote directory server.

For details, see "Understanding How Failures Are Handled" in the *Administration Guide*.

• Simplified Replication Server Setup

Use the new **setup replication-server** command to set up a server as a standalone replication server.

You can use this command to install standalone replication servers without specifying the base DNs to replicate.

For details, see "*Installing a Replication Server*" in the *Installation Guide*.

After preparing standalone replication servers, install directory servers as described in "*Installing a Directory Server*" in the *Installation Guide*. Configure the directory servers as replicas that use the replication servers. For details, see "Configuring Replication Settings" in the *Administration Guide*.

1.2.1. Product Improvements

DS 5.5.3

• There are no new product enhancements in this release, only bug fixes.

DS 5.5.2

• There are no new product enhancements in this release, only bug fixes.

DS 5.5.1

• There are no new product enhancements in this release, only bug fixes.

DS 5.5

This release of Directory Services software includes the following enhancement:

• Directory Proxy

  • The new proxy backend property, `heartbeat-search-request-base-dn`, lets you configure proxy backend heartbeat requests to target an entry under a base DN of interest rather than targeting the root DSE.

- When the global configuration property, `trust-transaction-ids`, is set to `true`, the proxy backend now adds a ForgeRock transaction ID control before forwarding the request, even if the incoming request did not include the control.

  As a result, all proxied requests have ForgeRock transaction IDs when you configure the server to trust transaction IDs.

- Monitoring

  - Replication server configurations now include these advanced properties for monitoring disk space use and stopping operations when the disk is full:

    disk-low-threshold

    > When this threshold is reached, the server logs warnings and sends warnings to the disk space monitoring subsystem.

    > The directory administrator must take action to provide more disk space.

    disk-full-threshold

    > When this threshold is reached, the server stops operations and lets connected directory servers fail over to another replication server. The replication server can resume operations once free disk space rises above the `disk-low-threshold` setting.

    For details, see "Replication Server" in the *Configuration Reference*.

  - Servers and the REST to LDAP gateway now include a ForgeRock transaction ID with each request.

    If you do not configure the server or gateway to trust transaction IDs in client application requests, then they ignore incoming transaction IDs, and instead generate a transaction ID for each request.

    If you configure the server or gateway to trust transaction IDs in client application requests, then outgoing requests reuse the incoming transaction ID. For each outgoing request in the transaction, the request's transaction ID has the form *original-transaction-id/sequence-number*, where *sequence-number* reflects the position of the request in the series of requests for this transaction. For example, if the *original-transaction-id* is `abc123`, the first outgoing request has transaction ID `abc123/0`, the second `abc123/1`, the third `abc123/2`, and so on. This helps you to distinguish specific requests within a transaction when correlating audit events from multiple services.

    To configure a server to trust transaction IDs in client application requests, set the global configuration property, `trust-transaction-ids`, to `true`.

    To configure the REST to LDAP gateway to trust transaction IDs in client application requests, set the JVM system property, `org.forgerock.http.TrustTransactionHeader`, to `true` in the web application container where the gateway runs.

3

- When an internal search is unindexed, a directory server now logs a message.

  For details, see "Determining Which Indexes Are Needed" in the *Administration Guide*.

- REST to LDAP

  - REST APIs now support the `_sortKeys` parameter to request that the server sort the query results it returns.

    For details, see "Server-Side Sort" in the *Developer's Guide*.

  - REST to LDAP now uses the affinity load balancer. This load balancer is described in "Choosing a Load Balancing Algorithm" in the *Administration Guide*.

- Security

  - OpenDJ servers use SHA-256 as the signature algorithm when generating key pairs, as an attacker with sufficient computing power could break SHA-1. NIST Special Publication 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, generally disallows use of SHA-1 for digital signature generation.

  - OpenDJ servers now support using a PKCS#11 device, such as a hardware security module (HSM), as a truststore.

    > **Note**
    >
    > When you use a PKCS#11 device as a truststore, all trusted certificates must be present on the device. No CA certificates are available by default. Import all the signing certificates required for your deployment before configuring the device for use as a truststore.

    To use an HSM as a truststore:

    1. Configure the JVM to allow access to the PKCS#11 device.

    2. Using the **dsconfig**, create an OpenDJ PKCS#11 Trust Manager Provider configuration to access the PKCS#11 device as a truststore.

- Tools

  - The **dsconfig** command now allows you to switch to advanced mode while using the command interactively.

  - The **setup --help** command now presents options in sorted order.

1.3. Security Advisories

ForgeRock issues security advisories in collaboration with our customers and the open source community to address any security vulnerabilities transparently and rapidly. ForgeRock's security advisory policy governs the process on how security issues are submitted, received, and evaluated as well as the timeline for the issuance of security advisories and patches.

For details of all the security advisories across ForgeRock products, see Security Advisories in the *Knowledge Base library*.

Chapter 2
Before You Install

This chapter covers requirements for running Directory Services software in production. It covers the following topics:

- Downloading Directory Services software

- Choosing hardware

- Choosing an operating system

- Preparing the Java environment

- Choosing an application server when using the DSML or REST to LDAP gateway

- Assigning FQDNs when using replication

- Using appropriately signed digital certificates

2.1. Downloading Directory Services Software

The ForgeRock BackStage site provides access to ForgeRock releases. ForgeRock releases are thoroughly validated for ForgeRock customers who run the software in production deployments, and for those who want to try or test a given release.

"Directory Services Software" describes the available software.

Directory Services Software

File	Description
`DS-5.5.3.zip`	Cross-platform distribution of the server software.
	Pure Java, high-performance server that can be configured as:
	• An LDAPv3 directory server with the additional capability to serve directory data to REST applications over HTTP.
	• An LDAPv3 directory proxy server providing a single point of access to underlying directory servers.
	• A replication server handling replication traffic with directory servers and with other replication servers, receiving and sending changes to directory data.

File	Description
	Server distributions include command-line tools for installing, configuring, and managing servers. The tools make it possible to script all operations. By default, this file unpacks into an `opendj/` directory.
`DS-5.5.3.msi`	Microsoft Windows native installer for the server software. By default, this installs files into a `C:\Program Files (x86)\OpenDJ\` directory.
`DS-5.5.3-1_all.deb`	Server software native packages for Debian and related Linux distributions. By default, this installs files into an `/opt/opendj/` directory.
`DS-5.5.3-1.noarch.rpm`	Server software native packages for Red Hat and related Linux distributions. By default, this installs files into an `/opt/opendj/` directory.
`DS-dsml-servlet-5.5.3.war`	Cross-platform DSML gateway web archive.
`DS-rest2ldap-servlet-5.5.3.war`	Cross-platform REST to LDAP gateway web archive.

2.2. Choosing Hardware

Thanks to the underlying Java platform, Directory Services software runs well on a variety of processor architectures. Many directory service deployments meet their service-level agreements without the very latest or very fastest hardware.

2.2.1. Fulfilling Memory Requirements

When installing a directory server for evaluation, you need 256 MB memory (32-bit) or 1 GB memory (64-bit) available, with 150 MB free disk space for the software and a small set of sample data.

For installation in production, read the rest of this section. You need at least 2 GB memory for a directory server and four times the disk space needed for initial production data in LDIF format. A replicated directory server stores data, indexes for the data, operational attribute data, and historical information for replication. The server configuration trades disk space for performance and resilience, compacting and purging data for good performance and for protection against temporary outages. In addition, leave space for growth in database size as client applications modify and add entries over time.

For a more accurate estimate of the disk space needed, import a known fraction of the initial LDIF with the server configured for production. Run tests to estimate change and growth in directory data, and extrapolate from the actual space occupied in testing to estimate the disk space required in production.

Directory servers almost always benefit from caching all directory database files in system memory. Reading from and writing to memory is much faster than reading from and writing to disk storage.

For large directories with millions of user directory entries, there might not be room to install enough memory to cache everything. To improve performance in such cases, use quality solid state drives either for all directory data, or as an intermediate cache between memory and disk storage.

2.2.2. Choosing a Processor Architecture

Processor architectures that provide fast single thread execution tend to help Directory Services software deliver the lowest response times. For top-end performance in terms of sub-millisecond response times and of throughput ranging from tens of thousands to hundreds of thousands of operations per second, the latest x86/x64 architecture chips tend to perform better than others.

Chip multi-threading (CMT) processors can work well for directory servers providing pure search throughput, though response times are higher. However, CMT processors are slow to absorb hundreds or thousands of write operations per second. Their slower threads get blocked waiting on resources, and thus are not optimal for deployments with high write throughput requirements.

2.2.3. Fulfilling Network Requirements

On systems with fast processors and enough memory to cache directory data completely, the network can become a bottleneck. Even if a single 1 Gbit Ethernet interface offers plenty of bandwidth to handle your average traffic load, it can be too small for peak traffic loads. Consider using separate interfaces for administrative traffic and for application traffic.

To estimate the network hardware required, calculate the size of the data returned to applications during peak load. For example, if you expect to have a peak load of 100,000 searches per second, each returning a full 8 KB entry, you require a network that can handle 800 MB/sec (3.2 Gbit/sec) throughput, not counting other operations, such as replication traffic.

2.2.4. Fulfilling Storage Requirements

> **Note**
>
> The directory server does not currently support network file systems such as NFS for database storage. Provide sufficient disk space on local storage such as internal disk or an attached disk array.

For a directory server, storage hardware must house both directory data, including historical data for replication, and server logs. On a heavily used server, you might improve performance by putting access logs on dedicated storage.

Storage must keep pace with throughput for write operations. Write throughput can arise from modify, modify DN, add, and delete operations, and from bind operations when a login timestamp is recorded, and when account lockout is configured, for example.

In a replicated topology, a directory server writes entries to disk when they are changed, and a replication server writes changelog entries. The server also records historical information to resolve potential replication conflicts.

As for network throughput, base storage throughput required on peak loads rather than average loads.

2.3. Choosing an Operating System

Directory Services 5.5 software is supported on the following operating systems:

- Linux 2.6 and later

- Microsoft Windows Server 2008, 2008 R2, 2012, 2012 R2, and 2016

In order to avoid directory database file corruption after crashes or power failures on Linux systems, enable file system write barriers and make sure that the file system journaling mode is ordered. For details on how to enable write barriers and how to set the journaling mode for data, see the options for your file system in the **mount** command manual page.

2.3.1. Setting Maximum Open Files

An OpenDJ server needs to be able to open many file descriptors, especially when handling thousands of client connections. Linux systems in particular often set a limit of 1024 per user, which is too low to handle many client connections to an OpenDJ server.

When setting up an OpenDJ server for production use, make sure the server can use at least 64K (65536) file descriptors. For example, when running the server as user `opendj` on a Linux system that uses `/etc/security/limits.conf` to set user level limits, you can set soft and hard limits by adding these lines to the file:

```
opendj soft nofile 65536
opendj hard nofile 131072
```

The example above assumes the system has enough file descriptors available overall. You can check the Linux system overall maximum as follows:

```
$ cat /proc/sys/fs/file-max
204252
```

2.3.2. Setting Maximum Inotify Watches

A directory server backend database monitors file events. On Linux systems, backend databases use the inotify API for this purpose. The kernel tunable `fs.inotify.max_user_watches` indicates the maximum number of files a user can watch with the inotify API. Make sure this tunable is set to at least 512K:

```
$ sysctl fs.inotify.max_user_watches
fs.inotify.max_user_watches = 524288
```

If this tunable is set lower than that, change it as shown in the following example:

```
$ sudo sysctl --write fs.inotify.max_user_watches=524288
[sudo] password for opendj:
fs.inotify.max_user_watches = 524288
```

2.3.3. Preventing Interference With Antivirus Software

Prevent antivirus and intrusion detection systems from interfering with OpenDJ software.

Before using OpenDJ software with antivirus or intrusion detection software, consider the following potential problems:

Interference with normal file access

Antivirus and intrusion detection systems that perform virus scanning, sweep scanning, or deep file inspection are not compatible with OpenDJ file access, particularly database file access.

Antivirus and intrusion detection software can interfere with the normal process of opening and closing database working files. They may incorrectly mark such files as suspect to infection due to normal database processing, which involves opening and closing files in line with the database's internal logic.

Prevent antivirus and intrusion detection systems from scanning database and changelog database files.

At minimum, configure antivirus software to whitelist the OpenDJ server database files. By default, exclude the following file system directories from virus scanning:

- `/path/to/opendj/changelogDb/` (if replication is enabled)

  Prevent the antivirus software from scanning these changelog database files.

- `/path/to/opendj/db/`

  Prevent the antivirus software from scanning database files, especially `*.jdb` files.

Port blocking

Antivirus and intrusion detection software can block ports that OpenDJ uses to provide directory services.

Make sure that your software does not block the ports that OpenDJ software uses. For details, see "Limiting System and Administrative Access" in the *Security Guide*.

Negative performance impact

Antivirus software consumes system resources, reducing resources available to other services including OpenDJ servers.

Running antivirus software can therefore have a significant negative impact on OpenDJ server performance. Make sure that you test and account for the performance impact of running antivirus software before deploying OpenDJ software on the same systems.

2.4. Preparing the Java Environment

Directory Services software consists of pure Java applications. Directory Services servers and clients run on any system with full Java support. Directory Services is tested on a variety of operating systems, and supported on those listed in "Choosing an Operating System".

Directory Services software requires Java 8, specifically at least the Java Standard Edition runtime environment, or the corresponding Java Development Kit to compile Java plugins and applications.

> **Note**
>
> ForgeRock validates Directory Services software with OpenJDK and Oracle JDK, and does occasionally run sanity tests with other JDKs such as the IBM JDK and Azul's Zulu. Support for very specific Java and hardware combinations is best-effort. This means that if you encounter an issue when using a particular JVM/hardware combination, you must also demonstrate the problem on a system that is widespread and easily tested by any member of the community.
>
> ForgeRock has tested this release and demonstrated that you can set up and run servers with Java 9. However, ForgeRock does not support running this release with Java 9.

ForgeRock recommends that you keep your Java installation up-to-date with the latest security fixes.

> **Important**
>
> Directory server JE database backends can require additional JVM options. When running a directory server with a 64-bit JVM and less than 32 GB maximum heap size, you must use the Java option, `-XX:+UseCompressedOops`. To use the option, edit the `config/java.properties` file. The following example settings include the option with the arguments for offline LDIF import, for rebuilding backend indexes, and for starting the directory server:
>
> ```
> import-ldif.offline.java-args=-server -XX:+UseCompressedOops
> rebuild-index.offline.java-args=-server -XX:+UseCompressedOops
> start-ds.java-args=-server -XX:+UseCompressedOops
> ```

Make sure you have a required Java environment installed on the system. If your default Java environment is not appropriate, set `OPENDJ_JAVA_HOME` to the path to the correct Java environment, or set `OPENDJ_JAVA_BIN` to the absolute path of the **java** command. The `OPENDJ_JAVA_BIN` environment variable is useful if you have both 32-bit and 64-bit versions of the Java environment installed, and want to make sure you use the 64-bit version.

2.5. Choosing an Application Server

OpenDJ servers run as standalone Java services, and do not depend on an application server.

The REST to LDAP and DSML gateway applications run on Apache Tomcat and Jetty.

ForgeRock supports only stable application container releases. See the Tomcat and Jetty documentation for details about the right container to use with your Java environment.

2.6. Assigning FQDNs For Replication

Directory Services replication requires use of fully qualified domain names, such as `opendj.example.com`.

Host names like `my-laptop.local` are acceptable for evaluation. In production, and when using replication across systems, you must either ensure DNS is set up correctly to provide fully qualified domain names, or update the hosts file (`/etc/hosts` or `C:\Windows\System32\drivers\etc\hosts`) to supply unique, fully qualified domain names.

2.7. Getting Digital Certificates Signed

If you plan to configure SSL or TLS to secure network communications between the server and client applications, install a properly signed digital certificate that your client applications recognize, such as one that works with your organization's PKI or one signed by a recognized certificate authority.

To use the certificate during installation, the certificate must be located in a file-based keystore supported by the JVM (JKS, JCEKS, PKCS#12), or on a PKCS#11 token. To import a signed certificate into a keystore, use the Java **keytool** command.

For details, see "Preparing For Secure Communications" in the *Administration Guide*.

2.8. Special Requests

If you have a special request regarding support for a combination not listed here, contact ForgeRock at info@forgerock.com.

Chapter 3
Compatibility

This chapter covers both major changes to existing functionality, and also deprecated and removed functionality.

3.1. Important Changes to Existing Functionality

Take the following changes into account when upgrading to Directory Services 5.5:

Important Changes in Directory Services 5.5.3

- No new changes to existing functionality were made in this release.

Important Changes in Directory Services 5.5.2

- No new changes to existing functionality were made in this release.

Important Changes in Directory Services 5.5.1

- No new changes to existing functionality were made in this release.

Important Changes in Directory Services 5.5

- The port-related options in "Options With No Default Values", and their short versions (such as `-p` for `--port`), no longer have default values when used in non-interactive mode.

Options With No Default Values

Commands Affected	Options Affected
addrate **authrate** **backup** **control-panel** **dsconfig** **export-ldif** **import-ldif** **ldapcompare** **ldapdelete**	`--port`

Commands Affected	Options Affected
ldapmodify **ldappasswordmodify** **ldapsearch** **manage-account** **manage-tasks** **modrate** **rebuild-index** **restore** **searchrate** **stop-ds**	

dsreplication	`--port`
	`--port1`
	`--port2`
	`--portDestination`
	`--portSource`
	`--replicationPort1`
	`--replicationPort2`

- The `-t | --numThreads` option for the following tools has changed to `-t | --numConcurrentRequests`:

  addrate
  authrate
  modrate
  searchrate

- The **dsreplication** command's `--baseDn` option is now only available where it is applicable.

  The **reset-change-number**, **resume**, **status**, and **suspend** subcommands no longer accept a `--baseDn` option.

- The product name has been aligned with the official name of the software release. The full product name starting with this release is `ForgeRock Directory Services`.

  This change impacts clients that depend on the product name.

  It also impacts the name used in product subcomponents. For example, in earlier releases the Syslog handler sent messages with the process name `OpenDJ`. The Syslog handler now sends messages with the process name `ForgeRock`.

- The server-side (plugin) Java API is continuing to evolve, as noted in "*Release Levels and Interface Stability*" in the *Reference*.

  Server plugins written against this API will have to be adapted and recompiled to work with this version. For Java API reference documentation, see the Javadoc.

- Manually changing the enabled property of an external change log domain will return incoherent results across the topology and as such is not supported.

3.2. Deprecated Functionality

This section lists deprecated functionality. Deprecation is defined in "ForgeRock Product Interface Stability" in the *Reference*.

Deprecated in Directory Services 5.5.3

• No new functionality was deprecated in this release.

Deprecated in Directory Services 5.5.2

• No new functionality was deprecated in this release.

Deprecated in Directory Services 5.5.1

• No new functionality was deprecated in this release.

Deprecated in Directory Services 5.5

• The PDB database backend type is deprecated and will be removed in a future release. Change your PDB backends to JE backends as described in "To Move a PDB Backend to a JE Backend" in the *Installation Guide*.

• The **dsreplication** subcommands **enable** and **disable** are deprecated and will be removed in a future release.

The subcommands have been replaced with **configure** and **unconfigure**, which more accurately reflect the permanence of the configuration changes made by these subcommands.

The **configure** subcommand updates the server configuration to replicate data under the specified base DN.

The **unconfigure** subcommand removes the replication configuration settings for the specified base DN, and removes references to the current server on other replicas.

The **dsreplication disable --disableAll** subcommand option is now **dsreplication unconfigure --unconfigureAll**. The **dsreplication disable --disableReplicationServer** subcommand option is now **dsreplication unconfigure --unconfigureReplicationServer**.

• The **control-panel** command is deprecated and will be removed in a future release.

• The configuration expression implementation is deprecated, and expected to change in a future release. This mechanism is described in "Using Configuration Expressions" in the *Administration Guide*.

3.3. Removed Functionality

Removed in Directory Services 5.5.3

• No new functionality was removed in this release.

Removed in Directory Services 5.5.2

• No new functionality was removed in this release.

Removed in Directory Services 5.5.1

• No new functionality was removed in this release.

Removed in Directory Services 5.5

• Support for Java 7 has been removed.

  Before upgrading to this version, you must follow the instructions in "Upgrading Java" in the *Installation Guide*.

• Support for Solaris has been removed.

• The **setup** command no longer supports addition of an `instance.loc` file to specify the instance path during server setup.

  If you do create an `instance.loc` file prior to setting up the server, the **setup** command fails with an error indicating either that the server has already been set up (when the `instance.loc` file references a valid server instance path), or that the `instance.loc` file (when the path it references does not exist, yet).

  Use the **setup** `--instancePath` option instead.

• The **uninstall** command has been removed.

  For details on removing server software, see "*Removing Server Software*" in the *Installation Guide*.

• The advanced JE backend properties, `db-evictor-lru-only` and `db-evictor-nodes-per-scan`, have been removed. When you upgrade a directory server, the **upgrade** command removes these properties from the configuration.

Chapter 4
Fixes, Limitations, and Known Issues

This chapter covers the status of key issues and limitations for Directory Services 5.5.

4.1. Key Fixes

This section covers key bug fixes in Directory Services 5.5 software.

Fixed Issues in DS 5.5.3

- OPENDJ-4625: Changelog range searches miss entries
- OPENDJ-5002: 200s timeout when stopping a replication server
- OPENDJ-5423: Incorrectly reported missing parent entries cause import-ldif and index rebuilds to fail
- OPENDJ-5553: Rest2Ldap cannot connect to TLSv1.2 servers
- OPENDJ-5582: LdapClientSocket connection leaked when handshake fails
- OPENDJ-5607: Update third party libraries
- OPENDJ-6377: Replication replay: issues with ReplaySynchronizer
- OPENDJ-6447: NullPointerException: formatString was null
- OPENDJ-6464: isMemberOfVirtualAttributeProvider does not process subordinate nested groups
- OPENDJ-6474: REST: some requests fail when stressing embedded http endpoint with Gatling

Fixed Issues in DS 5.5.2

- OPENDJ-4590: Replication: cursor aborted on high write throughput
- OPENDJ-4598: Replication Server cursoring through obsolete replica ID's causing high CPU spin
- OPENDJ-4934: Replication: changelog not in sync when restarting a server in a topology
- OPENDJ-4947: SASL DIGEST-MD5: bind request failed with protocol error
- OPENDJ-4983: IllegalStateException in change number indexer

- OPENDJ-5137: Reading compressed entries fails to close the InflaterInputStream

- OPENDJ-5210: Possible memory-leak if request received while bind in progress

- OPENDJ-5260: Grizzly pre-allocates a useless MemoryManager

- OPENDJ-5272: "idle-time-limit" global configuration property has no effect

- OPENDJ-5274: Upgrade of RS fails on rebuild indexes phase

- OPENDJ-5320: Build fails if the code is build outside the sustaining repo

Fixed Issues in DS 5.5.1

- OPENDJ-4624: Changelog search filter optimizations fail when there are leading unrelated terms

- OPENDJ-4295: Syslog data is not fully RFC compliant

Fixed Issues in DS 5.5.0

- OPENDJ-4341: setup with production mode with java 9

- OPENDJ-4316: HTTP Connector leaks Session objects

- OPENDJ-4275: Changelog searches cursor through inappropriate replica DBs

- OPENDJ-4234: Poor changelog search performance using changenumber ranges

- OPENDJ-4228: status command with keystore options throws ArrayIndexOutOfBoundsException

- OPENDJ-4178: Performance drop with complex subtree searches between 2.x and 3.5.1/4.0.0

- OPENDJ-4125: Extremely poor performance under connect/disconnect load and eventual port exhaustion

- OPENDJ-4115: build and publish missing changes gets confused with non-local changes

- OPENDJ-4011: Setup requires TLS to be enabled when using --productionMode

- OPENDJ-4007: Referential Integrity plugin checks all modifications when run as preModifyOperation

- OPENDJ-4006: forgerock-je included in releases does not work with Azul Zulu

- OPENDJ-3966: The Bcrypt storage scheme displays the wrong syntax Range and default for the bcrypt-cost

- OPENDJ-3963: JMXClientConnections are leaked

- OPENDJ-3931: Replication fails to propagate all changes added after a backup/restore to a newly created instance

- OPENDJ-3904: Delivery includes QuickSetup.app and Uninstall.app files for commands that were removed

- OPENDJ-3886: Modifying Json File-Based Access Logger configuration can cause a corrupt log record

- OPENDJ-3868: Proxied persistent searches are not cancelled/abandoned when the client abandons them or disconnects

- OPENDJ-3825: Spring daylight savings change can break recurring tasks

- OPENDJ-3645: SASL DIGEST-MD5: "digest-uri" parameter is not taken into account

- OPENDJ-3643: On Windows "java.properties" does not support values containing "=" character

- OPENDJ-3507: After upgrading a 2.6.2 server to 3.5.1 server is spinning at 93% CPU

- OPENDJ-3471: ldifsearch command fails to consume @objectclass notation in attribute list

- OPENDJ-3380: Creating a backend with null base DN can render the instance unusable

- OPENDJ-2850: OpenDJ SDK SASL integrity/confidentiality violates protocol

- OPENDJ-2842: Load balancing algorithms are not optimum after failure of a connection factory

- OPENDJ-2190: Replicas cannot always keep up with sustained high write throughput

- OPENDJ-1135: DS sometimes fails to connect to RS after server restart

- OPENDJ-609: Replicas out of sync after add/delete operations in sustained stress testing

4.2. Limitations

The following limitations exist in the following releases:

Limitations in Directory Services 5.5.3

- There are no new limitations in functionality in this release.

Limitations in Directory Services 5.5.2

- There are no new limitations in functionality in this release.

Limitations in Directory Services 5.5.1

- There are no limitations in functionality in this release.

Limitations in Directory Services 5.5

Directory Services 5.5 has the following limitations:

- Work around issues with hostname verification of local server certificates for administrative connections by doing the following:

  - When starting the Control Panel, select Remote Server and provide the hostname and administration port number, even when connecting to a local server.

  - When running the **status** command, use the `--trustAll` to bypass hostname verification.

- Configuring a server with both local backends and proxy backends is not supported.

  As described in "*Configuring Privileges and Access Control*" in the *Administration Guide*, access control models for directory servers and proxy servers cannot function at the same time in the same server.

- OpenDJ servers provide full LDAP v3 support, except for alias dereferencing, and limited support for LDAPv2.

- Directory servers store passwords prefixed with the storage scheme in braces, as in `{scheme}`. For details, see "Configuring Password Storage" in the *Administration Guide*.

  To prevent users from effectively attempting to choose their own password storage scheme, directory servers do not support passwords that strictly match this format. Specifically, directory servers do not support passwords that match `{string}*`.

  Requests to update `userPassword` values with such passwords fail with result code 19 (Constraint Violation) and an additional message indicating that passwords may not be provided in pre-encoded form.

- When you configure account lockout as part of password policy, an OpenDJ server locks an account after the specified number of consecutive authentication failures. Account lockout is not transactional across a replication topology, however. Global account lockout occurs as soon as the authentication failure times have been replicated.

- When configuring replication between servers of different versions, use the **dsreplication** command installed with the *newer* version.

  The **dsreplication enable** command in versions 3.5 and earlier is not compatible with Directory Services 5.5 and later servers.

- When creating additional database backends, adjust the database cache settings to avoid allocating all memory available to the JVM to database cache. Over-allocating memory to database cache leads to out of memory errors.

By default, a new database backend has `db-cache-percent` set to `50`. When creating a new database backend, you can raise or lower this value by using the `--set db-cache-percent:`*value* option, where *value* is the percentage of JVM memory to allocate to the new backend.

- The policy-based access control handler used in proxy servers:

  - Does not support the Get Effective Rights control.

  - Does not check the `modify-acl` privilege when global access control policies are changed. The `config-write` privilege is sufficient to change global access control policies.

  - Does not send alert notifications when global access control policies change.

- The Password Policy control (OID: `1.3.6.1.4.1.42.2.27.8.5.1`) is supported for add, bind, and modify operations. It is not supported for compare, delete, search and modify DN operations.

- Prevent antivirus and intrusion detection systems from interfering with OpenDJ software.

  Before using OpenDJ software with antivirus or intrusion detection software, consider the following potential problems:

  Interference with normal file access

  > Antivirus and intrusion detection systems that perform virus scanning, sweep scanning, or deep file inspection are not compatible with OpenDJ file access, particularly database file access.

  > Antivirus and intrusion detection software can interfere with the normal process of opening and closing database working files. They may incorrectly mark such files as suspect to infection due to normal database processing, which involves opening and closing files in line with the database's internal logic.

  > Prevent antivirus and intrusion detection systems from scanning database and changelog database files.

  > At minimum, configure antivirus software to whitelist the OpenDJ server database files. By default, exclude the following file system directories from virus scanning:

  > - `/path/to/opendj/changelogDb/` (if replication is enabled)

  >   Prevent the antivirus software from scanning these changelog database files.

  > - `/path/to/opendj/db/`

  >   Prevent the antivirus software from scanning database files, especially `*.jdb` files.

  Port blocking

  > Antivirus and intrusion detection software can block ports that OpenDJ uses to provide directory services.

Make sure that your software does not block the ports that OpenDJ software uses. For details, see "Limiting System and Administrative Access" in the *Security Guide*.

Negative performance impact

Antivirus software consumes system resources, reducing resources available to other services including OpenDJ servers.

Running antivirus software can therefore have a significant negative impact on OpenDJ server performance. Make sure that you test and account for the performance impact of running antivirus software before deploying OpenDJ software on the same systems.

- REST to LDAP query filters do not work with properties of subtypes.

  For example, the default example configuration describes a user type, and a POSIX user type that inherits from the user type. If your query filter is based on a POSIX user type property that is not a property of the user type, such as `loginShell` or `gidNumber`, the filter always evaluates to false, and the query returns nothing.

- When applying a Common REST patch operation, described in "Patching Resources" in the *Developer's Guide*, to a `Json` syntax attribute, you cannot patch individual fields of the JSON object. You must change the entire JSON object instead.

  As a workaround, you can perform an update of the entire object, changing only the desired fields in your copy.

- When the global server property `invalid-attribute-syntax-behavior` is set to `accept` or `warn`, a search on group membership using a value with invalid syntax returns nothing.

- Due to a Java issue on Windows systems (JDK-8057894), when configuring an OpenDJ directory server with data confidentiality enabled you might see an error message containing the following text:

```
Unexpected CryptoAPI failure generating seed
```

  If this happens, try running the command again.

- When running on a Linux or UNIX system without X11, the **status** command fails with an exception message such as the following:

```
Exception in thread "main" java.awt.AWTError: Can't connect to X11 window server using ':99' as the
  value of the DISPLAY variable.
```

  To work around this issue, set the Java property, `-Djava.awt.headless=true`.

4.3. Known Issues

> **Tip**
>
> When deploying OpenDJ servers in production, make sure that you follow the installation instructions. Allow OpenDJ servers to use at least 64K (65536) file descriptors. Also tune the JVM appropriately.

Known Issues in Directory Services 5.5.3

• There are no new known issues in this release.

Known Issues in Directory Services 5.5.2

• There are no new known issues in this release.

Known Issues in Directory Services 5.5.1

• There are no new known issues in this release.

Known Issues in Directory Services 5.5

The following important issues were known to exist in Directory Services 5.5.3:

• OPENDJ-4008: dsconfig exits with error when listing global access control policy

• OPENDJ-4059: dsconfig --bindDN should default to "cn=Directory Manager"

• OPENDJ-4106: Incorrect error when importing bad LDIF on setup

• OPENDJ-4109: The ldappasswordmodify command fails when requested through a directory proxy server

• OPENDJ-4185: Changelog not populated with new changes if an RS+DS goes down and replication fails to catch up when it's restarted

• OPENDJ-4226: Online list backups command throws error

• OPENDJ-4229: status command with keystore options throws NullPointerException

• OPENDJ-4243: Replication status's Age of Oldest Missing Change (AOMC) is not reset even if Missing Changes (MC) is 0

• OPENDJ-4312: addrate raises NoSuchElementException when using numusers

• OPENDJ-4325: Changelog searches requesting changelogCookie are very slow

• OPENDJ-4851: Exception when uninstalling/stopping replication topology

- OPENDJ-5070: Over allocation of db-cache-percent for existing backend results in empty error

- OPENDJ-5115: ldappasswordmodify fails, NPE in PasswordPolicyState updatePasswordHistory()

- OPENDJ-5140: PersistentSearch heap usage grows

- OPENDJ-5474: java.awt.AWTError when running status command on system without X11

Chapter 5
Documentation Updates

> **Warning**
>
> Many examples in the documentation trust server certificates with the `--trustAll` option.
>
> Examples using the `--trustAll` option are insecure except within a trusted network segment.
>
> In production deployments, use appropriate trust options. For details, see the Tools Reference in the *Reference*.

"Documentation Change Log" tracks important changes to the documentation:

Documentation Change Log

Date	Description
2020-11-06	Corrected the lists of key fixes for 5.5.0 and 5.5.1.
2020-04-03	Initial release of ForgeRock Directory Services 5.5.3. Added "To Disable Change Number Indexing" in the *Administration Guide* to explain how to disable change number indexing when not needed. For example, disable change number indexing when using DS as a CTS store for AM.
2018-11-26	The following documentation updates were made: • Added "*On Using a Load Balancer*" in the *Administration Guide* with recommendations for your deployment. • Updated "Preventing Interference With Antivirus Software" in the *Installation Guide* to clarify how to prevent interference.
2018-10-01	Release of ForgeRock Directory Services 5.5.2. The following documentation updates were made: • Removed the `-A` option in the **modrate** and **searchrate** tool reference (see Tools Reference in the *Reference*). This option was never supported.
2018-07-19	Release of ForgeRock Directory Services 5.5.1. The following documentation updates were made: • Clarified when you must use `-XX:+UseCompressedOops` in "Preparing the Java Environment". • Documented the impact of the name change from `OpenDJ` to `ForgeRock Directory Services` in "Important Changes to Existing Functionality".

Date	Description
	• Documented that each backend needs its own backup directory in "Backing Up Directory Data" in the *Administration Guide*.
	• Clarified that Solaris is no longer supported in "Removed Functionality".
	• Fixed contradictory tuning advice for JVM new generation in "Java Settings" in the *Administration Guide*.
	• Corrected the synopsis for `targattrfilters` in "ACI Targets" in the *Administration Guide*. The documentation incorrectly suggested `(targattrfilters != "`*expression*`")` as a legal ACI target. In an ACI target, `targattrfilters` must be set equal to an expression, as in `(targattrfilters = "`*expression*`")`.
	• Updated "Using Built-In Functions" in the *Administration Guide* to reflect that the `read()` and `readProperties()` functions require absolute paths, not relative paths.
	• Fixed the example for setting `group-id` on a replication server in "To Set Up Replication Groups" in the *Administration Guide*.
	• Fixed the explanation of how the REST to LDAP gateway maps an HTTP Basic user name to an LDAP bind DN in "Gateway Configuration File" in the *Reference*.
	• Fixed broken links to the Configuration Reference.
	• Sorted properties correctly in the Configuration Reference.
	• Fixed an error in the online LDIF export example shown in "To Export LDIF Data" in the *Administration Guide*.
	• The Javadoc now describes all ForgeRock classes and interfaces required to write server plugins and LDAP client applications.
2017-10-18	Initial release of ForgeRock Directory Services 5.5. In addition to the changes described in "*What's New*" and "*Compatibility*", the following important changes were made to the documentation:
	• A new LDAP Schema Reference is available. This reference covers the LDAP schema elements defined by default.
	• "Database Cache Settings" in the *Administration Guide* now better describes caching for larger directories, such as those with 10 million entries or more.
	• "Initializing Replicas" in the *Administration Guide* has been updated to suggest how to initialize replication depending on your circumstances.
	• The documentation on *Breaking a Multi-Role Server Into Standalone Components* has been removed.

FORGEROCK

Date	Description
	Instead of creating a multi-role (DS/RS) server and then splitting it apart, use the **setup** command to create standalone servers at installation time. For details, see "*Installing a Directory Server*" in the *Installation Guide* and "*Installing a Replication Server*" in the *Installation Guide*.

27

Chapter 6
Getting Support

This chapter offers information and resources about Directory Services and ForgeRock support.

6.1. Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

- The ForgeRock Knowledge Base offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

  While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock product documentation, such as this document, aims to be technically accurate and complete with respect to the software documented. It is visible to everyone and covers all product features and examples of how to use them.

6.2. Using the ForgeRock.org Site

The ForgeRock.org site has links to source code for ForgeRock open source software, as well as links to the ForgeRock forums and technical blogs.

If you are a *ForgeRock customer*, raise a support ticket instead of using the forums. ForgeRock support professionals will get in touch to help you.

6.3. How to Report Problems and Provide Feedback

If you have questions regarding Directory Services software that are not answered by the documentation, you can ask questions on the OpenDJ forum under https://forgerock.org/forum/fr-projects/opendj/.

When requesting help with a problem, include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation

- Description of the environment, including the following information:

- Machine type

- Operating system and version

- Storage type and version

- Java version

- Web container and version (if applicable)

- Directory Services release version

- Any patches or other software that might be affecting the problem

- Steps to reproduce the problem

- Any relevant access and error logs, stack traces, or core dumps

6.4. Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, training through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. For details on ForgeRock's support offering, including support plans and service level agreements (SLAs), visit https://www.forgerock.com/support.