Release Notes

ForgeRock Directory Services 5

Mark Craig

ForgeRock AS
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

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

Table of Contents

About ForgeRock Directory Services Software iv
1. What's New 1
    1.1. New Features 1
    1.2. Product Improvements 3
2. Before You Install 6
    2.1. Downloading ForgeRock Directory Services Software 6
    2.2. Choosing Hardware 7
    2.3. Choosing an Operating System 9
    2.4. Preparing the Java Environment 10
    2.5. Choosing an Application Server 10
    2.6. Assigning FQDNs For Replication 11
    2.7. Getting Digital Certificates Signed 11
    2.8. Special Requests 11
3. Compatibility 12
    3.1. Important Changes to Existing Functionality 12
    3.2. Deprecated Functionality 20
    3.3. Removed Functionality 20
4. Fixes, Limitations, and Known Issues 22
    4.1. Key Fixes 22
    4.2. Limitations 24
    4.3. Known Issues 26
5. Documentation Updates 29
6. Getting Support 31
    6.1. Accessing Documentation Online 31
    6.2. Joining the ForgeRock Community 31
    6.3. How to Report Problems and Provide Feedback 32
    6.4. Getting Support and Contacting ForgeRock 32

About ForgeRock Directory Services Software

ForgeRock Identity Platform™ is the only offering for access management, identity management, user-managed access, directory services, and an identity gateway, designed and built as a single, unified platform.

The platform includes the following components that extend what is available in open source projects to provide fully featured, enterprise-ready software:

- ForgeRock Access Management (AM)
- ForgeRock Identity Management (IDM)
- ForgeRock Directory Services (DS)
- ForgeRock Identity Gateway (IG)

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

ForgeRock 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 ForgeRock 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. ForgeRock Directory Services software comes with plenty of tools. ForgeRock Directory Services software also offers REST access to directory data over HTTP.

ForgeRock 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 ForgeRock 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 ForgeRock Directory Services software.

Chapter 1
What's New

This chapter covers new capabilities in ForgeRock Directory Services 5.

1.1. New Features

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

LDAP Directory Proxy Services

OpenDJ server software now offers LDAP directory proxy services.

LDAP directory proxy services make it easier to deploy:

- A single point of access providing a uniform view of underlying LDAP directory services.
- High service availability, hiding implementation details from LDAP client applications.
- LDAP load balancing and failover strategies to handle referrals, connection failures, and network partitions.

For details, see Chapter 4, "*Installing a Directory Proxy Server*" in the *Installation Guide* and Chapter 15, "*Configuring LDAP Proxy Services*" in the *Administration Guide*.

When you set up a directory proxy server, access control is implemented using global access control policy entries, rather than global ACIs. For more information about global access control policies, see Section 6.5, "About Global Access Control Policies" in the *Administration Guide*.

Security By Default

OpenDJ server configurations can now be hardened for production environments during installation using the **setup** command option, `--productionMode`.

In a hardened configuration, server features are set appropriately for secure production deployments, including especially the following features:

- Global access control is appropriately restricted.
- Default password policies use stronger password storage and prevent use of common passwords.
- Transport layer security protocols and cipher suites for securing connections are restricted to more secure options.

- Data confidentiality (encryption) is enabled for database backends and default indexes.
- Log permissions on UNIX/Linux systems prevent other members of the same group from reading the logs.

For additional details, see Section 6.7, "Set Up Servers in Production Mode" in the *Security Guide*.

New Common Audit Handlers

OpenDJ servers now support new ForgeRock Common Audit event handlers for:

- JSON files as described in Section 18.5.1.1.1, "Configuring JSON Access Logs" in the *Administration Guide*.

  This format is now the default for server access logs.
- Java Message Service (JMS) as described in Section 18.5.1.1.5, "Configuring JMS Access Logs" in the *Administration Guide*.
- Splunk as described in Section 18.5.1.1.6, "Configuring Splunk Access Logs" in the *Administration Guide*.

In addition, you can now find example Common Audit-related configuration files under `config/audit-handlers/`.

JSON Syntax for LDAP

OpenDJ servers now support LDAP attribute values that have JSON syntax. This makes it possible to index JSON values and to search for them using ForgeRock Common REST query filter expressions within LDAP search filters.

For details, see Example 4.10, "Search: Using JSON Query Filters" in the *Developer's Guide*, Example 4.19, "Modify: Updating a JSON Syntax Attribute" in the *Developer's Guide*, and Example 7.4, "Configuring an Index for a JSON Attribute" in the *Administration Guide*.

When using REST to LDAP, you can map these attributes with the `json` type described in Table A.2, "Properties of Resource Type Properties Objects" in the *Reference*.

LDAP-Based Keystore

OpenDJ servers now implement an `OpenDJ` security provider for LDAP and LDIF-based keystore services.

For examples showing how to use the implementation, see Section 7.3, "Using an LDAP Keystore" in the *Security Guide*.

EL Configuration Expressions

OpenDJ server configuration files now support EL expressions, making it possible to use variables in configuration files.

For details, see Section 2.4, "Using Configuration Expressions" in the *Administration Guide*.

REST API Documentation

OpenDJ servers now provide OpenAPI (formerly Swagger) descriptors for REST APIs.

For details on preparing and publishing REST API reference documentation that your developers can use, see Section 3.12, "Working With REST API Documentation" in the *Developer's Guide*.

Windows Native Packaging

Native Windows packages are now available for OpenDJ server software.

For details, see the Installation Guide.

1.2. Product Improvements

This release of ForgeRock Directory Services software includes the following enhancements:

API for Embedding OpenDJ Servers

This release includes an improved API and examples that demonstrate embedding OpenDJ server software in your application.

For details, see Chapter 10, "*Embedding the Server*" in the *Developer's Guide*.

REST/HTTP Data Access

Base DN references in REST to LDAP configurations now support templates and `..` notation.

For details, see Table A.2, "Properties of Resource Type Properties Objects" in the *Reference*.

Root DSE Configuration Improvement

When the Root DSE backend configuration property, `show-subordinate-naming-contexts`, is set to true, the root DSE exposes sub-suffix naming contexts separately.

By default, only top-level naming contexts are visible.

Security Improvements

- All tools now fully support TLSv1.2-only deployments.

  OpenDJ server software already supported TLSv1.2-only deployments.

- OpenDJ server software now has documented support for PKCS#11 tokens, including hardware security modules.

  For details, see Section 7.4, "Using a Hardware Security Module" in the *Security Guide*.

- The server administration connector has new properties, `allowed-client` and `denied-client`.

These properties let you specify a set of host names or address masks to determine which clients can and cannot establish administrative connections.

Simpler Security Configuration

When you configure a component, such as a connection handler, that relies on key manager providers and trust manager providers, you can now let the server use the JVM settings. OpenDJ servers expose a JVM Key Manager and a JVM Trust Manager.

These providers inherit their settings from the JVM configuration. If you change the JVM security configuration, restart the server to inherit the new configuration.

By default, the JVM, and therefore the JVM Key Manager, does not specify access to private keys. The JVM Key Manager can be useful, for example, when keys are stored in a hardware security module or other keystore, and the JVM is configured to provide system-wide access to the server keys.

The JVM Trust Manager uses the JVM's truststore, `$JAVA_HOME/jre/lib/security/cacerts` by default. This truststore contains many well-known CA certificates.

The JVM does provide a default truststore, `$JAVA_HOME/jre/lib/security/cacerts`, for validating well-known CA certificates. By default, this is the truststore used by JVM Trust Manager Provider.

Tools Improvements

- OpenDJ client and server command-line tools now share the same implementation and interfaces.

  The server tools now include the following performance testing commands:

  - **addrate**
  - **authrate**
  - **modrate**
  - **searchrate**

  The **ldif-diff** command is now **ldifdiff**. The **make-ldif** command is now **makeldif**.

  A number of command-line options have been added or changed. For details regarding interface changes, see Table 3.1, "Changes To Command-Line Tools".

- The **dsjavaproperties** command is now no longer necessary and has been removed. You can simply update `config/java.properties` and restart the tool or the server for the changes to take effect.

- The **makeldif** command now gzips LDIF output if the output filename ends in `.gz`.

- Templates for the **makeldif** command now support wrapped lines.

- The **dsreplication** command includes new subcommands, **suspend** and **resume**.

  For examples, see Procedure 8.6, "To Stop Replication Temporarily For a Replica" in the *Administration Guide*.

Chapter 2
Before You Install

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

- Downloading ForgeRock 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 ForgeRock 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.

Table 2.1, "ForgeRock Directory Services Software" describes the available software.

Table 2.1. ForgeRock Directory Services Software

File	Description
DS-5.0.0.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-5.0.0.msi`	Microsoft Windows native installer for the server software. By default, this installs files into a `C:\Program Files (x86)\OpenDJ\` directory.
`DS-5.0.0-1_all.deb`	Server software native packages for Debian and related Linux distributions. By default, this installs files into an `/opt/opendj/` directory.
`DS-5.0.0-1.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-5.0.0.war`	Cross-platform DSML gateway web archive.
`DS-rest2ldap-servlet-5.0.0.war`	Cross-platform REST to LDAP gateway web archive.

The platform version number that appears in the download file names may differ from the internal version number. The internal version number for this release is 4.0.0.

2.2. Choosing Hardware

Thanks to the underlying Java platform, ForgeRock 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 an 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 ForgeRock 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

ForgeRock Directory Services 5 software is supported on the following operating systems:

- Linux 2.6 and later
- Microsoft Windows Server 2008, 2008 R2, 2012, and 2012 R2
- Oracle Solaris 10, 11

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. Preventing Interference With Antivirus Software

Prevent antivirus and intrusion detection systems from interfering with directory services.

Antivirus and intrusion detection systems that do a deep inspection of database files are not compatible with OpenDJ server software. Disable antivirus and intrusion detection systems, or at least prevent them from operating on OpenDJ server files.

2.4. Preparing the Java Environment

ForgeRock Directory Services software consists of pure Java applications. ForgeRock Directory Services servers and clients run on any system with full Java support. ForgeRock Directory Services is tested on a variety of operating systems, and supported on those listed in Section 2.3, "Choosing an Operating System".

ForgeRock Directory Services software requires Java 7 or 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 ForgeRock 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 recommends that you keep your Java installation up-to-date with the latest security fixes.

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

ForgeRock 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 Section 5.2, "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 ForgeRock Directory Services 5:

- Commands delivered with OpenDJ server software have changed. Some commands have new options. Commands that output LDIF now do not wrap the LDIF by default. (You can change this by using the new `-t, --wrapColumn {int}` option.) Interfaces for some commands have changed, as described in Table 3.1, "Changes To Command-Line Tools".

Table 3.1. Changes To Command-Line Tools

Tool	Changes
backup	When running the command with the server stopped, the command now requires the `--offline` option.
dsjavaproperties	The command has been removed. After you update `config/java.properties`, restart the server or run the command-line tool again to use the new Java settings.
export-ldif	When running the command with the server stopped, the command now requires the `--offline` option.
import-ldif	The `--skipDNValidation` option is no longer available. DN validation is now always performed as part of the second import phase. This improves overall import rate when all DNs are valid. Invalid DNs are detected, however, only after the first pass through the LDIF to import. As a result, problems with invalid DNs are found later in the process. If you suspect that some entries in the LDIF might be invalid, use the `--rejectFile` option to capture entries rejected by the server during import. When running the command with the server stopped, the command now requires the `--offline` option.
ldapcompare	The synopsis has changed. The command now requires a single DN as an argument: `ldapcompare {options} attribute:value DN`

Tool	Changes
	Invoke the command multiple times to compare multiple entries. The following options have been removed: • `-c, --continueOnError` • `-f, --filename` (for multiple comparisons, use the command multiple times) • `-i, --encoding {encoding}` • `-r, --useSASLExternal` (use `-o, --saslOption mech="EXTERNAL"` instead) • `-s, --script-friendly` (renamed `-S, --scriptFriendly`) • `-V, --ldapVersion {version}` (the command always binds according to LDAPv3) The default value for the `-h, --hostname` was previously `localhost`. To allow SSL hostname verification, it now defaults to the host FQDN. When the comparison is false, the command returns 5, whether or not you use the `-m, --useReturnCode` option.
ldapdelete	The synopsis has changed. The command now optionally accepts a single DN as a trailing argument, or reads one or more DNs on separate lines from standard input: `ldapdelete {options} [DN]` The following options have been removed: • `-f, --filename` (read from standard input instead) • `-i, --encoding {encoding}` • `-r, --useSASLExternal` (use `-o, --saslOption mech="EXTERNAL"` instead) • `-V, --ldapVersion {version}` (the command always binds according to LDAPv3) The default value for the `-h, --hostname` was previously `localhost`. To allow SSL hostname verification, it now defaults to the host FQDN.
ldapmodify	The synopsis has changed. The command now optionally accepts one or more LDIF change files as trailing arguments, or reads LDIF from standard input: `ldapmodify {options} [changes.ldif ...]` The following options have been removed: • `-a, --defaultAdd` (now default) • `-f, --filename` (still available for compatibility, but hidden and not recommended) • `-i, --encoding {encoding}`

Tool	Changes
	• `-r, --useSASLExternal` (use `-o, --saslOption mech="EXTERNAL"` instead) • `-V, --ldapVersion {version}` (the command always binds according to LDAPv3) The default value for the `-h, --hostname` was previously `localhost`. To allow SSL hostname verification, it now defaults to the host FQDN.
ldappasswordmodify	The following options have been removed: • `-A, --provideDNForAuthzID` (now default) • `-N` (use long option `--newPasswordFile` instead) The default value for the `-h, --hostname` was previously `localhost`. To allow SSL hostname verification, it now defaults to the host FQDN.
ldapsearch	The synopsis has changed. The filter argument is now mandatory: `ldapsearch {options} filter [attributes ...]` The following options have been removed: • `-c, --continueOnError` • `-f, --filename` • `-i, --encoding {encoding}` • `-r, --useSASLExternal` (use `-o, --saslOption mech="EXTERNAL"` instead) • `-T, --dontWrap` (use the new option `-t, --wrapColumn` instead) • `-V, --ldapVersion {version}` (the command always binds according to LDAPv3) The default value for the `-h, --hostname` was previously `localhost`. To allow SSL hostname verification, it now defaults to the host FQDN. The `-s, --searchScope {searchScope}` now takes a plural, `subordinates`, to request the subordinate subtree search scope.
ldif-diff	This command has been renamed **ldifdiff**. The synopsis now resembles that of the **diff** command, where the source and target LDIF are mandatory trailing arguments: `ldifdiff [options] source.ldif target.ldif` You can also provide the source and target LDIF using standard input, as described in Section 4.3.4, "Using Standard Input With the LDIF Tools" in the *Administration Guide*. The following options have been removed: • `-a, --ignoreAttrs {file}`

Tool	Changes
	Use `-e, --excludeAttribute {attribute}` instead. • `--ignoreEntries {file}` Use `-B, --excludeBranch {branchDN}` instead. • `-O, --overwriteExisting` (now default) • `-r, --useCompareResultCode` • `-s, --sourceLDIF {file}` (use the mandatory argument instead) • `-S, --singleValueChanges` • `-t, --targetLDIF {file}` (use the mandatory argument instead) • `--checkSchema` The command now returns 0 if no differences are found, and returns 1 if differences are found. If you do not use or do not supply a filename argument to the `-o, --outputLDIF {ldifFile}` option, the command writes the results on standard output.
ldifmodify	The synopsis has changed. The source file is now required as a trailing argument. The command now optionally accepts one or more LDIF change files as trailing arguments, or reads LDIF from standard input: `ldifmodify {options} source.ldif changes.ldif [changes.ldif ...]` You can also provide the source and changes LDIF using standard input, as described in Section 4.3.4, "Using Standard Input With the LDIF Tools" in the *Administration Guide*. The following options have been removed: • `-s, --sourceLDIF {file}` (now a mandatory trailing argument) • `-m, --changesLDIF {ldifFile}` (use trailing arguments or standard input) • `-t, --targetLDIF {ldifFile}` (use the option `-o, --outputLDIF` instead) If you do not use or do not supply a filename argument to the `-o, --outputLDIF {ldifFile}` option, the command writes the results on standard output.
ldifsearch	The synopsis has changed. The filter argument is now mandatory: `ldifsearch {options} source.ldif filter [attributes ...]` You can also provide the source LDIF using standard input, as described in Section 4.3.4, "Using Standard Input With the LDIF Tools" in the *Administration Guide*. The `-b, --baseDN` option can now only be used once.

Tool	Changes
	The following options have been removed: • `-f, --f {filterFileName}` • `-l, --ldifFile {ldifFile}` (now a mandatory trailing argument) • `-O, --overwriteExisting` (now default) • `-t` (now the short option for `--wrapColumn`; use the long option `--timeLimit` instead) • `-T, --dontWrap` (use the new option `-t, --wrapColumn` instead) • `-o, --outputFile` (renamed `--outputLDIF`) If you do not use or do not supply a filename argument to the `-o, --outputLDIF {ldifFile}` option, the command writes the results on standard output.
list-backends	The command has been removed. Use the **dsconfig list-backends** command instead, or read the data sources output of the **status** command.
make-ldif	The command has been renamed **makeldif**. The synopsis has changed. The command now requires a template file path as a trailing argument: `makeldif {options} templateFile` The following options have been removed: • `-t, --templateFile` (now a mandatory trailing argument) • `--ldifFile` (renamed `--outputLDIF`)
rebuild-index	When running the command with the server stopped, the command now requires the `--offline` option.
restore	When running the command with the server stopped, the command now requires the `--offline` option.
setup	The command has changed significantly. Update your installation procedures to align with the following changes: • The GUI installer is no longer available. The **setup** command is a command-line only tool. • The **setup** command now optionally takes a subcommand. This makes it possible to install the software as a directory server, as a standalone replication server, or as a directory proxy server. • Interactive and silent modes are now fully distinct. Either you start the command without arguments for interactive mode, or you supply all mandatory

Tool	Changes
	arguments for silent mode. The **setup** command no longer supports specifying some arguments and providing others interactively. • The **setup** command fails if the server has already been configured. The current command assumes a server has already been configured if it has a `config` directory or a `db` directory. • Default keystore and truststore configuration has been simplified. When setting up a new server, the command uses a single keystore. This keystore holds private keys and public key certificates. It functions as the keystore and truststore for establishing secure communication for administration connections, for HTTPS, for LDAP with StartTLS, and for LDAPS. If you provide a keystore at setup time, the command uses that keystore for these purposes. If you explicitly request a generated key pair and self-signed certificate, or if you do not specify security parameters, the command generates a file-based keystore, `config/keystore`. The default format is PKCS#12. The generated key pair has a self-signed certificate with alias `server-cert`. The generated password for the private key and the keystore is stored in the protected file `config/keystore.pin`. • The **setup** command no longer supports configuring the JMX connection handler. Configure JMX as described in Section 5.10, "JMX Client Access" in the *Administration Guide* instead. • When used non-interactively, the **setup** command no longer supplies default values, except the default root user DN, `cn=Directory Manager`, and the database type. (The **setup** command still suggests default values when used interactively.) In non-interactive mode, you must now supply the host name and port numbers. If you plan to use replication or secure connections with remote systems, the host name should be an FQDN, such as `opendj.example.com`. Conventional port numbers are listed in Table D.1, "Server Ports" in the *Reference*. • The **setup** command no longer supports using a properties file to specify arguments. Supply all necessary arguments when running the command instead. The following options are mandatory.

Tool	Changes
	If you use only these options, the command sets up a server listening only on an administration port. The administration port is protected by a key pair generated at setup time with a self-signed certificate: • `--adminConnectorPort {port}` (conventional port number: 4444) • `--hostname {hostname}` • `--rootUserDN {rootUserDN}` (default: `cn=Directory Manager`) • `--rootUserPassword {rootUserPassword}` The following options have been added: • `--httpPort {port}` (enables HTTP access using the HTTP connection handler) • `--httpsPort {port}` (enables HTTPS access using the HTTP connection handler) • `--instancePath {path}` (recommended way to install the server software separately from the server configuration, logs, and data files) The **setup** command still supports use of an `instance.loc` file as before. The following options have been removed: • `--generateSelfSignedCertificate` • `-i, --cli` • `--jmxPort {jmxPort}` • `-n, --no-prompt` • `--noPropertiesFile` • `--propertiesFilePath {propertiesFilePath}` • `--verbose`

The way truststores are used by these commands has changed. For details on the current behavior, see Section 4.1.2, "How Command-Line Tools Trust Server Certificates" in the *Developer's Guide*.

- For newly installed servers, the default password policies now use stronger password storage schemes:
  - The default password policy for normal users now uses the Salted SHA-512 password storage scheme instead of the Salted SHA-1 storage scheme.
  - The default password policy for root DN users now uses the PBKDF2 password storage scheme instead of the Salted SHA-512 storage scheme.

In addition, new root DN user passwords must be at least 8 characters in length.

This change does not affect upgraded servers.

- When specifying a branch in **makeldif** templates, you must now also specify the object classes for the branch. For example, suppose a template creates an organizational unit branch as follows:

```
branch: ou=People,[suffix]
```

You now create the organization unit branch as follows:

```
branch: ou=People,[suffix]
objectClass: top
objectClass: organizationalUnit
```

For details on writing **makeldif** templates, see makeldif.template(5) in the *Reference*.

- For fresh installations of an OpenDJ server, the JSON-based LDAP access logger is now the default. For details, see Section 18.5.1.1.1, "Configuring JSON Access Logs" in the *Administration Guide*.

  The previous logger is still available, but is no longer enabled by default for new servers. For details on enabling the native LDAP access logger, see Section 18.5.1.2, "Native LDAP Access Logs" in the *Administration Guide*.

  When you upgrade a server, its log configuration does not change.

- Due to internal changes in request processing, the `etime` values in access logs now include the time spent waiting for a worker thread to start processing the request. Previously, the timer began only when the worker thread started to process the operation.

- The `index-entry-limit` property is now marked as an advanced property. To view `index-entry-limit` settings, use the **dsconfig** `--advanced` option.

  Before changing the value of the `index-entry-limit` property, read Section 7.4.4, "Understanding Index Entry Limits" in the *Administration Guide*.

- The server-side (plugin) Java API is continuing to evolve, as noted in Appendix I, "*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 Server Javadoc.

- The location to put optional additional .jar files that are required for your deployment and that are not delivered with the server has changed. Now use *`instance-path`*`/extlib/`, as described in Appendix C, "*File Layout*" in the *Reference*.

3.2. Deprecated Functionality

This section lists deprecated functionality. Deprecation is defined in Section I.2, "ForgeRock Product Interface Stability" in the *Reference*.

- 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 Procedure 9.4, "To Move a PDB Backend to a JE Backend" in the *Installation Guide*.
- Support for Java 7 is deprecated and will be removed in the next 5.5 release.

  When upgrading to the current release, also move to Java 8 in order to be prepared for pending removal of support for Java 7.
- 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.
- Using a `instance.loc` file to specify the instance path during server setup is deprecated. This feature will be removed in a future release.

  Use the **setup** `--instancePath` option instead.
- The **uninstall** command is deprecated and will be removed in a future release.

  Stop the server and remove files instead.

3.3. Removed Functionality

- The **dsjavaproperties** command has been removed.
- The **list-backends** command has been removed.
- Previously deprecated environment variables beginning with `OPENDS` are no longer supported.

Use `OPENDJ_JAVA_BIN` and `OPENDJ_JAVA_ARGS` instead.

- The advanced global configuration property, `server-error-result-code`, has been removed.

  The result code used for internal server errors is the LDAP Other error code, 80.

- The advanced backend database property, `preload-time-limit`, is not supported in this release.

Chapter 4

Fixes, Limitations, and Known Issues

This chapter covers the status of key issues and limitations for ForgeRock Directory Services 5.

4.1. Key Fixes

The following important bugs were fixed in this release:

- OPENDJ-3650: Modify-increment values are rejected as not being integers
- OPENDJ-3488: Removing an Auxiliary Objectclass from a user in a replicated topology not applied on the remote server
- OPENDJ-3456: JE growth in OpenAM site config with embedded DJ deployment
- OPENDJ-3446: ZipException during backup results in a failed backup when a duplicate log entry is found
- OPENDJ-3445: When the LDAP port is not accessible, ds-cfg-symmetric-key values are not being replicated correctly
- OPENDJ-3428: JE cleaner threads stop deleting files
- OPENDJ-3380: Creating a backend with null base DN can render the instance unusable
- OPENDJ-3375: dsconfig create-backend-index allows adding an index definition for non-valid schema names
- OPENDJ-3337: dsreplication status on a DS shows a DS+RS missing after the DS+RS is disabled/enabled
- OPENDJ-3309: Replication server connection listener thread exits silently
- OPENDJ-3288: Upgrading backends with compressed entries results in unusable JE backends
- OPENDJ-3283: Cleaner threads unable to clean files, changelogDb grows until disk fills up
- OPENDJ-3281: Modify operations may not be replayed if case is mixed on attribute values
- OPENDJ-3272: ClassCastException on creating password storage scheme via command line
- OPENDJ-3252: Enum Syntax: No such syntax is configured for use

- OPENDJ-3237: Disk full scenario can result in empty offline.state files and lead to changelogDb read failure
- OPENDJ-3231: dsreplication status uses wrong bind DN
- OPENDJ-3230: upgrade: running verify-index on objectclass index incorrectly reports errors
- OPENDJ-3223: upgrade to 3.5.0 should rebuild indexes using DN syntax
- OPENDJ-3221: dsconfig cannot connect when the Administration Connector is configured for TLSv1.2 only
- OPENDJ-3205: Control-panel: missing java-settings and manage tasks screen
- OPENDJ-3204: REST to LDAP gateway: container sometimes fails to stop when CTS resolver is configured
- OPENDJ-3203: Control-panel: creating a new base DN does not complete
- OPENDJ-3160: REST to LDAP reference property mappers do not support subresources
- OPENDJ-3147: Regressions on Virtual Static Group membership checks
- OPENDJ-3133: dsreplication status reports M.C. (Missing Changes) when none exist.
- OPENDJ-3098: Cannot configure syslog audit event logger
- OPENDJ-3055: Enabling fractional replication feature breaks replication
- OPENDJ-3034: Equality filter with an invalid attribute value evaluates as unindexed rather than an empty result set
- OPENDJ-3032: throwIfIA5IllegalCharacter does not check the first character
- OPENDJ-3000: Password Expiration notification calculation has integer overflow problem.
- OPENDJ-2976: Allow access log filtering for rootDSE searches.
- OPENDJ-2969: changelogDb could not be read on OpenDJ instance startup
- OPENDJ-2965: isMemberOf searches are inefficient
- OPENDJ-2963: subtreeSpecification filters using isMemberOf are inefficient
- OPENDJ-2858: Unable to parse LDIF record as ChangeRecord if empty attribute is listed first.
- OPENDJ-2846: PromptingTrustManager does not handle wildcard certificates correctly
- OPENDJ-2833: With invalid-attribute-syntax-behavior set, adding invalid syntax values to groupofuniquenames generates errors in log

- OPENDJ-2814: Invalid attribute syntax behavior fails to reject non-boolean syntax values
- OPENDJ-2801: Upgrade fails when the global setting "smtp-server" is missing the optional port
- OPENDJ-2748: dsconfig --batch and --batchFilePath fail when configuring the global-aci.
- OPENDJ-2738: DN validation fails when RDN uses a custom attribute
- OPENDJ-2731: Middle and final substring indexes fail to return candidates, resulting in an unindexed search.
- OPENDJ-2727: Low performance during import with large index-entry-limit
- OPENDJ-2721: JE is using all the available heap memory during import.
- OPENDJ-2719: PDB entries cannot be larger than 4MB
- OPENDJ-2697: Upgrading JE backend with mixed case loses data
- OPENDJ-2669: Incorrect messages for memory settings error
- OPENDJ-2659: Privileges can be lost after the BIND
- OPENDJ-2640: Online import doesn't delete temporary files after import completion.
- OPENDJ-2631: OOME error while importing 100M entries (online-import) causes the server to crash
- OPENDJ-2609: NoSuchElementException on ldapsearch --sortorder when using corresponding VLV index
- OPENDJ-2515: Common Audit throughput regression
- OPENDJ-2446: dsreplication purge-historical uses an inappropriate amount of server memory if many entries match search criteria
- OPENDJ-1976: setup.bat doesn't work without 8.3 format
- OPENDJ-1906: Improve static group refresh performance
- OPENDJ-1667: dsconfig batch file processing removes double and single-quotes from attribute values
- OPENDJ-1633: Unable to run tools in offline mode when tools.properties is set up
- OPENDJ-347: Misleading error when running setup

4.2. Limitations

This release has the following limitations:

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

  As described in Chapter 6, "*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.

- 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 ForgeRock Directory Services 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.

- Antivirus and intrusion detection systems that do a deep inspection of database files are not compatible with OpenDJ server software. Disable antivirus and intrusion detection systems, or at least prevent them from operating on OpenDJ server files.

- 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 Section 3.8, "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.

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.

The following important issues remained open at the time of this release:

- OPENDJ-3904: "QuickSetup.app" and "Uninstall.app" files should be removed from the delivery
- OPENDJ-3886: Modifying Json File-Based Access Logger may cause a corrupt log record
- OPENDJ-3868: Proxied persistent searches are not cancelled/abandoned when the client abandons them or disconnects
- OPENDJ-3825: Seemingly intermittent scheduling backup task error "nonexistent calendar date"
- OPENDJ-3706: Change number indexer's cursor can be aborted due to purge activity
- OPENDJ-3697: OPENDJ service using net start returns early with START_PENDING if OpenDJ starts slowly
- OPENDJ-3645: SASL DIGEST-MD5: "digest-uri" parameter is not taken into account
- OPENDJ-3614: Fully disabling replication using --hostname <IP> only disables the local instance
- OPENDJ-3609: ldif-diff/ldifdiff fails to properly differentiate schema files.
- OPENDJ-3579: Setting Logfile permissions with dsconfig has no effect on Windows

- OPENDJ-3507: After upgrading a 2.6.2 server to 3.5.1 server is spinning at 93% CPU
- OPENDJ-3494: PDB backend spins and runs out of memory if system clock is set backwards
- OPENDJ-3480: Updating schema backend properties while it's enabled leaves the backend in broken state
- OPENDJ-3471: ldifsearch no longer supports objectclass @ notation for attribute list
- OPENDJ-3469: Clicking Runtime Options - Java Settings results in an InvocationTargetException exception
- OPENDJ-3438: Online rebuild-index memory calculation is inappropriate when multiple PDB backends are involved
- OPENDJ-3437: Cannot delete access log publisher when it is disabled
- OPENDJ-3435: Paging controls ignored for certain query filters.
- OPENDJ-3427: PDB: cleanup_manager error after modrate
- OPENDJ-3410: Control-Panel: manage schema -> modifying a custom entry does not work
- OPENDJ-3406: dsreplication status hangs when client uses TLSv1.2 and server uses TLSv1.1
- OPENDJ-3399: DirectoryException while rebuilding index on JE instance during upgrade
- OPENDJ-3343: Invalid Conflict resolution on Add sequence when Parent & Child are added on different replica
- OPENDJ-3341: REST to LDAP gateway: HTTP response for API description is empty
- OPENDJ-3299: Editing an existing custom objectClass throws a ConflictingSchemaElementException exception
- OPENDJ-3291: PDB: TXN_UPDATE update thread CPU usage
- OPENDJ-3234: Unhelpful error messages when server cannot read/write tasks backend
- OPENDJ-3224: Infinite loop reading replication changelog if a CSN appears more than once
- OPENDJ-3212: java.lang.OutOfMemoryError occurred during upgrade
- OPENDJ-3182: PDB: CorruptJournalException while restoring backend
- OPENDJ-3153: REST to LDAP gateway: changing password fails when using proxied authorization
- OPENDJ-3070: JE backends corrupt when low on disk space
- OPENDJ-3057: Replication Server starts listener although ChangeLog DB is unusable
- OPENDJ-3054: ldapmodify silently discards duplicate values

- OPENDJ-3029: dsreplication disable --disableAll does not remove all replication data from other instances' cn=admin data backend.
- OPENDJ-2784: Modify RDN does not work if there ACIs with targetattrfilters deny(write) on ldap:/// anyone

Chapter 5
Documentation Updates

Table 5.1, "Documentation Change Log" tracks important changes to the documentation:

Table 5.1. Documentation Change Log

Date	Description
2017-04-13	Updated Section 18.5.1.1, "Common ForgeRock Access Logs" in the *Administration Guide* to indicate where to find ForgeRock Common Audit sample configuration samples. Refreshed release notes.
2017-03-29	Initial release of ForgeRock Directory Services 5. In addition to the new documentation mentioned in Chapter 1, "*What's New*", and changes described in Chapter 3, "*Compatibility*", the following important changes were made to the documentation: • A new guide to securing directory services is available, the Security Guide. • The Installation Guide has been reorganized to account for changes to the **setup** command, and to make instructions for each component more self-contained. • For information on resolving conflicts that cannot be resolved automatically during data replication, see Section 8.5, "Resolving Replication Conflicts" in the *Administration Guide*. • To set up a password policy in the spirit of recent NIST recommendations, see Procedure 10.2, "To Configure the Default Policy to Meet NIST Requirements" in the *Administration Guide*. • To limit the security protocols that command-line tools accept, see Procedure 7.15, "To Restrict Protocols For Command-Line Tools" in the *Security Guide*. • To better understand how database backends handle files on disk, see Section 4.4, "About Database Backends" in the *Administration Guide*. • To learn how to prove that confidential data is indeed encrypted, see Section 4.7, "Encrypting Directory Data" in the *Administration Guide*. • For examples demonstrating how the attribute value password validator works, see Procedure 10.1, "To Adjust the Default Password Policy" in the *Administration Guide*.

Date	Description

	• All chapters previously forming the *Directory Server Developer's Guide* have moved to the Developer's Guide. This includes examples and explanations for the following topics:
	  • Accessing directory data over HTTP using REST APIs
	  • Accessing directory data over LDAP using OpenDJ client tools
	  • Understanding and extending LDAP schema
	  • Working with groups of entries
	  • Working with virtual and collective attributes
	  • Working with referrals
	  The *Directory Server Developer's Guide* has been removed.
	• The reference for **dsconfig** subcommands has been moved to the *Server Configuration Reference*.

Chapter 6
Getting Support

This chapter offers information and resources about ForgeRock 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.
- ForgeRock core 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.

Core documentation therefore follows a three-phase review process designed to eliminate errors:

- Product managers and software architects review project documentation design with respect to the readers' software lifecycle needs.
- Subject matter experts review proposed documentation changes for technical accuracy and completeness with respect to the corresponding software.
- Quality experts validate implemented documentation changes for technical accuracy, completeness in scope, and usability for the readership.

The review process helps to ensure that documentation published for a ForgeRock release is technically accurate and complete.

Fully reviewed, published core documentation is available at http://backstage.forgerock.com/. Use this documentation when working with a ForgeRock Identity Platform release.

6.2. Joining the ForgeRock Community

Visit the Community resource center where you can find information about each project, download trial builds, browse the resource catalog, ask and answer questions on the forums, find community events near you, and find the source code for open source software.

6.3. How to Report Problems and Provide Feedback

If you have questions regarding ForgeRock 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)
  - ForgeRock 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, classes 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, visit https://www.forgerock.com, or send an email to ForgeRock at info@forgerock.com.