Release Notes

/ Directory Services 6

Latest update: 6.0.0

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 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
- Definitions for release levels and interface stability
- Getting support

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 6.

1.1. New Features

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

Backend Database Storage

- Directory servers now use an optimized JE backend implementation whose dependencies are distributed under the Apache License, Version 2.0. This license is suitable for all deployments, including OEM deployments.

  Important

  Support for PDB backend databases has been removed in this release.

  Before upgrading a directory server using any PDB backends, take one of the following actions:

  - Move the data to JE backend databases before upgrading.
  - Export all data in PDB backend databases to LDIF before upgrading, and import the data into the JE backend databases with the same names after upgrading.

- Backend indexes now have new time to live (TTL) properties to configure automated, optimized entry expiration and removal:

  - `ttl-enabled`
  - `ttl-age`

  For details on how to use this feature, see "Automating Entry Expiration and Deletion" in the *Administration Guide*.

- This release upgrades JE backend databases to Berkeley DB Java Edition 7.5.11.

Configuration Expressions

Server configuration expressions have been reimplemented to align with other ForgeRock Identity Platform™ software. Configuration expressions make it possible to substitute configuration property values with variables that you can set before starting DS servers.

For details, see "Using Configuration Property Value Substitution" in the *Administration Guide*.

Faster Bulk Updates

The **ldapmodify** and **ldapdelete** commands now offer a `--numConnections` option to perform updates in parallel on multiple LDAP connections.

This feature enables faster bulk updates, and provides an alternative to the **import-ldif --append** option removed in version 3.0.

For an example, see "Bulk Adding Entries" in the *Developer's Guide*.

Monitoring

DS server monitoring has been reimplemented to align with other ForgeRock Identity Platform™ software.

This change affects LDAP and HTTP monitoring interfaces, but not SNMP interfaces. The SNMP interface continues to use standard metrics.

Monitoring capabilities now have interface stability Stable as described in "ForgeRock Product Interface Stability".

For documentation about available interfaces and metrics, see "*Monitoring, Logging, and Alerts*" in the *Administration Guide* and "*Monitoring Metrics*" in the *Reference*.

DS servers now feature the following monitoring capabilities:

- Support for pulling monitoring data to Prometheus monitoring software.

  For an example, see "HTTP-Based Monitoring" in the *Administration Guide*.

- Support for pushing monitoring data to a Graphite service.

  For details, see "Monitoring With Graphite" in the *Administration Guide*.

- Support for creating a directory monitoring account during setup.

  For an example, see "To Set Up a Directory Server" in the *Installation Guide*.

  When using JMX to monitor the server with this account, be aware that in this release the account does not have JMX-related privileges. Instead, you must add the required JMX-related privileges as described in "To Configure Access To JMX" in the *Administration Guide*.

1.2. Product Improvements

This release of Directory Services software includes the following enhancements:

Backend Database Storage

This release improves many JE backend settings:

- A new advanced property, `db-durability`, makes the backend durability setting easier to configure and to read.

  For details, see "About Database Backends" in the *Administration Guide*.

- New defaults for disk space thresholds better fit modern deployments. The property, `disk-low-threshold`, defaults to 5% of the filesystem size, plus 5 GB. The property, `disk-full-threshold`, defaults to 5% of the filesystem size, plus 1 GB.

  For details, see "Setting Disk Space Thresholds For Database Backends" in the *Administration Guide*.

- Default limits for backend database log files better fit larger deployments. The property, `db-log-file-max`, defaults to 1 GB instead of 100 MB. The property, `db-log-filecache-size`, defaults to 200 instead of 100. As a result, the database can grow to 200 GB instead of 10 GB on disk before the file cache begins to close some database log files in order to open others.

  For details, see "Database Cache Settings" in the *Administration Guide*.

- The new properties, `db-run-log-verifier`, and `db-log-verifier-schedule`, make it possible to configure whether and when the server runs checksum verification on backend database logs.

  For details, see "About Database Backends" in the *Administration Guide*.

JSON Support

In addition to optimized indexes for JSON attribute values that are queried with ForgeRock® Common REST query filters, DS directory servers now also support optimizations for JSON with optional fields.

For details, see "Working With JSON" in the *Administration Guide*.

Monitoring

DS servers support a new privilege, `monitor-read`. This prevents unauthorized users from reading monitoring metrics unless they have the privilege.

Assign this privilege to users who read monitoring metrics over LDAP or HTTP. For an example, see "LDAP-Based Monitoring" in the *Administration Guide*.

When upgrading, see "To Upgrade Replicated Servers" in the *Installation Guide*, which shows how to add missing privileges to the global administrator account. These privileges are required when using the **dsreplication status** command.

Performance

This release of DS software includes many performance improvements.

No DS server performance improvements require action on your part, though optimal tuning settings including JVM settings may now be different for specific scenarios.

When upgrading to this release, be aware that the command-line performance tools have a new template value syntax as described in "Important Changes to Existing Functionality".

Replication

- DS servers now allow you to choose a single, global server ID used when configuring replication, rather than letting replication configuration randomly assign server IDs.

  Before configuring replication, you can set the global configuration property `server-id`.

  This makes it easier to keep track of server IDs when reviewing replication configuration, and to parameterize replication configuration in DevOps deployments.

  For an example, see "Configuring Replication" in the *Administration Guide*.

- DS directory servers store historical replication information for internal use in entries' `ds-sync-hist` attributes. This release introduces a new encoding that significantly reduces the space required to store `ds-sync-hist` data.

  The space reduction trades a smaller footprint for increased CPU use when preparing to write `ds-sync-hist` values. Read and search operations should not be negatively impacted, however. Indeed, read and search performance should improve to the extent that reduced entry size means more efficient use of backend database and CPU caches.

- DS replication domains can now be disabled by setting the configuration property, `enabled`, to `false`. If a replication domain is disabled, its contents are not replicated.

  This change facilitates parameterizing whether backends and associated replication domains are enabled, which is useful in DevOps deployments where not all environments replicate the same data on each replica.

  To suspend and later resume replication, see "To Stop Replication Temporarily For a Replica" in the *Administration Guide*.

  To disable replication, see "To Stop Replication Permanently For a Replica" in the *Administration Guide*.

Security

DS certificate mappers now support certificate issuer verification. Use this to verify the certificate issuer whenever multiple CAs are trusted in order to prevent impersonation. Different CAs can issue certificates with the same subject DN, but not with the same issuer DN.

This feature relies on the certificate mapper property, `issuer-attribute`, to identify the attribute in the user entry that holds the issuer DN. For this purpose, DS servers define the attribute, `"ds-certificate-issuer-dn"`, as an optional attribute of the `ds-certificate-user` object class. For an example using this attribute, see "Authenticating Client Applications With a Certificate" in the *Developer's Guide*.

Server Configuration

The **dsconfig** command now allows you to configure DS servers that are not running.

This release introduces an `--offline` option and a `--configFile` option. When you use the **dsconfig --offline** command with a DS server that is stopped, you change the server configuration file in the default location, such as `/path/to/opendj/config/config.ldif`. The `--configFile` option allows you to specify an alternative configuration file in offline mode.

Server-Side Sorting

DS servers now support the following improvements to server-side sorting:

- JSON ordering matching rules for JSON attributes.

  For details, see "Working With JSON" in the *Administration Guide*.

  DS servers can implement JSON ordering matching rules on demand. This enables REST to LDAP to support for `_sortKeys` where a field is inside the value of a JSON attribute.

- An extension of the server-side sort request sort order specification.

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

- A means for REST to LDAP to communicate the `_sortKeys` field that is inside the value of a JSON attribute to the DS server.

  This functionality is part of REST to LDAP. By using the `useServerSideSortForJson` boolean configuration parameter, you can configure whether to sort results in the REST to LDAP gateway, For details, see "Gateway REST2LDAP Configuration File" in the *Reference*.

- A means for the REST to LDAP gateway to limit the maximum number of entries supported by the local sort mechanism when sorting results based on JSON attributes.

  The setting is `localSortMaxEntries`.

  For details, see "Gateway REST2LDAP Configuration File" in the *Reference*.

Unindexed Searches

DS directory servers now support the following improvements for unindexed searches. These improvements are designed to help applications that store data in the directory as arbitrary JSON objects, and that provide a graphical UI for browsing directory data accessed over REST. With these improvements, users can page through directory data, sorting on whichever JSON field they choose without initially specifying any filter.

As with any unindexed search that you allow, the trade off is inefficient use of system resources and less performance. This is not, therefore, a general capability that should be provided to all applications without taking the impact into consideration. It is intended for use by a directory data administrator who is browsing data without knowing in advance what they are looking for:

- DS directory servers can now use an appropriately configured VLV index to sort results for an unindexed search.

  For details, see "VLV Index for Paged Server-Side Sort" in the *Administration Guide*.

- DS directory servers sort unindexed search results as long as they are paged.

  This improvement has the following limitations:

  - The simple paged results control must specify a *page size* that is less than or equal to the `index-entry-limit` (default: 4000).

  - For each page, the server reads the entire backend database, retaining *page size* number of sorted entries.

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
- Synchronizing System Clocks For 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-6.0.0.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.

File	Description
	• A replication server handling replication traffic with directory servers and with other replication servers, receiving and sending changes to directory data. 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-6.0.0.msi`	Microsoft Windows native installer for the server software. By default, this installs files into a `C:\Program Files (x86)\OpenDJ\` directory.
`DS-6.0.0-1_all.deb`	Server software native packages for Debian and related Linux distributions. By default, this installs files into an `/opt/opendj/` directory.
`DS-6.0.0-1.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-6.0.0.war`	Cross-platform DSML gateway web archive.
`DS-rest2ldap-servlet-6.0.0.war`	Cross-platform REST to LDAP gateway web archive.
`DS-monitoring-dashboard-samples-6.0.0.zip`	Sample Grafana dashboard demonstrating how to graph DS server metrics stored in a Prometheus database. You are responsible for adapting the sample to suit your production requirements. These resources are provided for *demonstration purposes only*. Commercial support for the ForgeRock DevOps Examples is not available from ForgeRock. For details on how to try the sample dashboard, see the `README.md` file delivered inside the .zip file.

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.

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. Fulfilling Minimum Disk Space Requirements

To evaluate DS software, make sure you have 10 GB free disk space for the software and for sample data.

The more data you have, the more disk space you need. Before deploying production systems, make sure you have enough space. For details, see "Planning for High Scale" in the *Deployment Guide*.

2.2.3. 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.

When deploying DS servers with replication enabled, allow at minimum two CPU cores per server. Allow more CPU cores per server, especially in high-volume deployments or when using CPU-intensive features such as encryption. Single CPU systems seriously limit server performance.

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.

2.2.4. 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 Gb 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 Gb/sec) throughput, not counting other operations, such as replication traffic.

2.2.5. 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 6 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

DS servers need 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 the DS server.

When setting up your DS 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 DS software.

Before using DS 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 DS 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 DS 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 DS uses to provide directory services.

Make sure that your software does not block the ports that DS 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 DS servers.

Running antivirus software can therefore have a significant negative impact on DS server performance. Make sure that you test and account for the performance impact of running antivirus software before deploying DS 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 or 9, 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 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. Running in a Container

For some settings, DS servers depend on system information reported by the JVM to determine defaults. When running DS servers in containers such as Docker, the JVM may return information about the operating system that does not reflect container constraints and limits. Unless you use a version of the JVM that supports gathering container information, as described in JDK-8146115, manually adjust the settings described below.

Before adjusting settings, determine the following container constraints:

- The number of CPU core hardware threads dedicated to the containerized system, which is usually twice the number of CPU cores
- The amount of RAM dedicated to the containerized system

When running DS servers in containers such as Docker, adjust the following settings:

- `num-request-handlers`

  Recommendation: Set this either to 2 or to 1/4 of the number of core hardware threads, whichever is larger.

- `num-worker-threads`

  Recommendation: Set this either to 4 or to 5/8 of the number of core hardware threads, whichever is larger.

- `db-num-cleaner-threads`

Recommendation: Set this either to 2 or to 1/4 of the number of core hardware threads, whichever is larger.

- `num-update-replay-threads`

  Recommendation: Set this either to 4 or to 1/2 of the number of core hardware threads, whichever is larger.

- `-Xmx` (Java setting limiting maximum heap size)

  To use the option, edit the `config/java.properties` file and restart the server.

  For example, consider a container limited to 8 GB RAM. The following setting limits the maximum heap size to 8 GB when starting the directory server:

  ```
  start-ds.java-args=-server -Xmx8G
  ```

- `db-cache-percent`

  If the directory server has multiple database backends, the total percent of JVM heap used must remain less than 100 (percent), and must leave space for other uses.

- `db-cache-size`

  The same rules apply when using this alternative to `db-cache-percent`. If you set its value larger than 0, then it takes precedence over `db-cache-percent`. Total JVM heap used must remain smaller than available RAM, and must leave space for other uses.

2.6. Choosing an Application Server

DS 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 (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.

2.7. Assigning FQDNs For Replication

Directory Services replication requires use of fully qualified domain names (FQDNs), 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 FQDNs, or update the hosts file (`/etc/hosts` or `C:\Windows\System32\drivers\etc\hosts`) to supply unique, FQDNs.

2.8. Synchronizing System Clocks For Replication

When using DS replication, keep server system clocks synchronized.

To keep the system clocks synchronized, use a tool that always moves the clock forwards. For example, `ntpd` adjusts the size of a second so that time always moves forwards to eventual clock consistency.

Never move the system clock *backwards*. Never use tools such as **ntpdate** that may move the clock backwards.

2.9. 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*.

2.10. 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 6:

- Root DN users no longer belong to a special group or have alternate names, nor are their accounts stored in the configuration file, `config.ldif`. Instead, directory superusers, such as `cn=Directory Manager`, are now stored in their own, separate backends whose base DN is the user DN.

  When you upgrade a server, the upgrade process moves existing root DN users to their own LDIF backends. The LDIF files for these backends are found in the `/path/to/opendj/db` directory.

  You can choose to store directory superuser entries in database backends instead of LDIF backends. This allows you to encrypt the data on disk, for example. (Recreate the backend as a JE backend, and then import from the LDIF file.) You can also create directory superusers manually as described in "To Use a Non-Default Superuser Account" in the *Security Guide*.

  Previously, root DN user profiles had an `alternate-bind-dn` property. This was used to allow you to specify bind DNs such as `cn=Directory Manager` instead of `cn=Directory Manager,cn=Root DNs,cn=config`. As the root user DNs are now top-level DNs, this mechanism is no longer supported.

  Directory superuser privileges are now specified as `ds-privilege-name` values on their entries.

  Also as a result of this change, the **dsconfig get-root-dn-prop** and **dsconfig set-root-dn-prop** subcommands are no longer supported.

- For new installations, defaults have changed for the following JE backend properties:

  - The default for `db-log-file-max` has increased from 100 MB to 1 GB.
  - The default for `db-log-filecache-size` has increased from 100 to 200.
  - The default for `disk-low-threshold` is now 5% of the filesystem size, plus 5 GB.
  - The default for `disk-full-threshold` is now 5% of the filesystem size, plus 1 GB.

  The new defaults for `disk-low-threshold` and `disk-full-threshold` apply for replication servers as well.

- Default connection handler names have been shortened. The "Connection Handler" suffixes have been dropped from the names. For example, the default "LDAP Connection Handler" is now named LDAP.
- Server configuration expressions have been reimplemented to align with other ForgeRock Identity Platform™ software.

  For details, see "Using Configuration Property Value Substitution" in the *Administration Guide*.
- The **setup** command option `--useJceks` has been renamed to `--useJceKeyStore`.

  The **setup** command option `--useJceksTrustStore` has been renamed to `--useJceTrustStore`.
- When creating a schema provider for a customized JSON query matching rule, the type to create is now `json-query-equality-matching-rule`, rather than `json-schema`.

  For details, see "Example: Custom Index Using a JSON Query Matching Rule" in the *Administration Guide*.
- The server-side (plugin) Java API is continuing to evolve, as noted in "*Release Levels and Interface Stability*".

  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.
- For new DS server installations, the file layout has changed to mutable data, which is changed by the server at runtime, from potentially immutable configuration data.

  When you *upgrade an existing server*, the following files remain where they were in the old layout:

  - LDAP schema files located in the `config/schema/` directory
  - The `config/ads-truststore.pin` and `config/ads-truststore.pin` files

  When you set up a new server, the new file layout is used for all files. The file names in the following table indicate where files have moved.

Old Layout	New Layout

`config/admin-backend.ldif`
 	
`config/admin-backend.ldif.old`
	`db/admin/admin-backend.ldif`
	
	`db/admin/admin-backend.ldif.old`

`config/ads-truststore`
 	
`config/ads-truststore.pin`
	`db/ads-truststore/ads-truststore`
	
	`db/ads-truststore/ads-truststore.pin`

`config/archived-configs`	`var/archived-configs`
`config/config.ldif.startok`	`var/config.ldif.startok`
All LDAP schema files that were in the `config/schema/` directory...	...are now in the `db/schema/` directory.

Old Layout	New Layout
`config/tasks.ldif`	`db/tasks/tasks.ldif`
All files that were in the `config/upgrade/` directory...	...are now in the `var/upgrade/` directory.

The new file layout is described in "*File Layout*" in the *Reference*.

- The command-line performance tools no longer accept **printf**-style format strings in templates. Instead, they use a `{1}`, `{2}`, `{n}` token syntax, where the `{1}` represents the first data source, `{2}` the second, and so on.

  As an example, the following command measures search throughput and response time. For each search, the command substitutes a random value for `{1}` from the specified range of `rand(0,2000)`:

  ```
  $  searchrate -p 1389 -b "dc=example,dc=com" -g "rand(0,2000)" "(uid=user.{1})"
  ```

  The tools also support relative indexing, using `{}` tokens without numbers. In the example above, `"(uid=user.{})"` would reference the `-g "rand(0,2000)"` data source.

  This change affects the following tools:

  addrate(1)
  authrate(1)
  modrate(1)
  searchrate(1)

3.2. Deprecated Functionality

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

- The HTTP monitoring endpoint, `/admin/monitor`, has been deprecated.

  Use `/metrics/api` or `/metrics/prometheus` instead.

- The output of the **status** command has been deprecated. Its content is expected to change significantly in a future release.

- The metrics for M.C. (missing changes) and A.O.M.C. (age of oldest missing change) shown by the **dsreplication status** command have been deprecated, and are likely to be removed in a future release.

  The following related metrics are deprecated as well:

  - `ds-mon-approx-oldest-change-not-synchronized` (LDAP)
  - `ds-mon-approximate-delay` (LDAP)

- `ds-mon-missing-changes` (LDAP)
- `ds_replication_changelog_connected_replicas_approx_oldest_change_not_synchronized_seconds` (Prometheus)
- `ds_replication_changelog_connected_replicas_approximate_delay_seconds` (Prometheus)
- `ds_replication_changelog_connected_replicas_missing_changes` (Prometheus)

Monitor replication delay instead. For details, see "Monitoring Replication Delay Over LDAP" in the *Administration Guide* or "Monitoring Replication Delay Over HTTP" in the *Administration Guide*.

3.3. Removed Functionality

- Support for Solaris has been removed.
- The control panel has been removed in this release. Use the command line tools instead.
- The **dsreplication** subcommands **enable** and **disable** have been removed in this release.

  Use the **configure** and **unconfigure** subcommands instead.
- Support for PDB backend databases has been removed in this release. This release supports JE backend databases.

  As a result, the **setup directory-server** option, `-t | --backendType`, has been removed.
- The JE backend database advanced properties, `db-txn-no-sync` and `db-txn-write-no-sync`, have been removed.

  Use `db-durability` instead.
- The EL expression implementation for using variables in server configurations has been removed in this release.

  Instead, use the implementation described in "Using Configuration Property Value Substitution" in the *Administration Guide*.
- The PIN and password related configuration properties listed in the following table have been removed.

Old Properties	Use This Instead...

`key-store-pin-environment-variable`
 	
`key-store-pin-file`
 	
`key-store-pin-property`
	`key-store-pin`

Old Properties	Use This Instead...

`trust-store-pin-environment-variable`
`trust-store-pin-file`
`trust-store-pin-property`
	`trust-store-pin`

`mapped-search-bind-password-environment-variable`
`mapped-search-bind-password-file`
`mapped-search-bind-password-property`
	`mapped-search-bind-password`

`proxy-user-password-environment-variable`
`proxy-user-password-file`
`proxy-user-password-property`
	`proxy-user-password`

`bind-password-environment-variable`
`bind-password-file`
`bind-password-property`
	`bind-password`

To replace these properties, use configuration expressions described in "Using Configuration Property Value Substitution" in the *Administration Guide*. For example, to replace `key-store-pin-file: config/keystore`, use `key-store-pin: &{file:config/keystore}`. To replace `key-store-pin-property: ds.keystore.pin`, use `key-store-pin: &{ds.keystore.pin}`.

- The **dsconfig get-root-dn-prop** and **dsconfig set-root-dn-prop** subcommands have been removed in this release.
- Support for assured replication has been removed in this release.

  The interface stability of assured replication has been classified as Internal.

Chapter 4

Fixes, Limitations, and Known Issues

This chapter covers the status of key issues and limitations for DS 6.

4.1. Key Fixes

The following important bugs were fixed in this release:

- OPENDJ-4823: Adding a third replica breaks key ordering of the changelogDb
- OPENDJ-4598: Replication Server cursoring through obsolete replica ID's causing high CPU spin
- OPENDJ-4587: Replication: Medium consistency point frozen when a DS+RS is unconfigured or a DS+RS is stopped
- OPENDJ-4555: Server not responding
- OPENDJ-4983: IllegalStateException in change number indexer
- OPENDJ-4210: Cannot import/export LDIF in offline mode after configuring OpenDJ Password Synchronization Plugin
- OPENDJ-4125: Extremely poor performance under connect/disconnect load and eventual port exhaustion
- OPENDJ-4729: WorkerThread is blocked in BlockingBackpressureOperator after disconnection
- OPENDJ-4485: MODRDN with a blank newrdn: value is not rejected.
- OPENDJ-4296: Rebuilding index on two backends at the same time causes NPE
- OPENDJ-4557: isMemberOf search result excludes entries' operational attributes
- OPENDJ-3437: Cannot delete access log publisher when it is disabled
- OPENDJ-4725: Cannot reset change-log change number
- OPENDJ-4845: Crypto manager uses TLSv1, fails if admin connector ssl-protocol is TLSv1.2
- OPENDJ-4943: NullPointerException in BackupManager.java when backup --hash is used offline
- OPENDJ-4559: All worker threads blocked on ReentrantReadWriteLock in GroupManager

- OPENDJ-3504: LDAP bytesRead/Written and SNMP counters (dsApplIfInBytes and dsApplIfOutBytes) are not incremented
- OPENDJ-4497: ttl-enabling an index requires a restart
- OPENDJ-4533: NullPointerException in TTL reaper
- OPENDJ-3878: Example plugin POM has wrong parent and is missing repositories
- OPENDJ-1881: OpenDJ JMX monitoring report statistics as type String instead of Number
- OPENDJ-3896: Change number indexer exits due to uncaught IllegalStateException
- OPENDJ-4464: Collective attributes do not consider if an attribute is single or multi-valued.
- OPENDJ-431: Server-side sort control only works on result sets of less than 100000 entries
- OPENDJ-934: Changes to RS window-size property require a server restart
- OPENDJ-1158: rebuild-index leaves backend offline if a backup is running

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 "*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.

- DS 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, DS directory servers lock 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 6 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 DS software.

  Before using DS 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 DS 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 DS 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 DS uses to provide directory services.

Make sure that your software does not block the ports that DS 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 DS servers.

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

- REST to LDAP does not support modify RDN operations.
- 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 DS directory servers 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 DS servers in production, make sure that you follow the installation instructions. Allow DS 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-4185: Changelog not populated with new changes if an RS+DS goes down and replication fails to catch up when it's restarted
- OPENDJ-4243: Replication status's Age of Oldest Missing Change (AOMC) is not reset even if Missing Changes (MC) is 0
- OPENDJ-4229: status command with keystore options throws NullPointerException
- OPENDJ-4764: REST2LDAP gateway sasl-plain authorization doesn't handle dn: correctly
- OPENDJ-4775: Proxy keeps searching on ports removed from Static Discovery Mechanism
- OPENDJ-4008: dsconfig exits with error when listing global access control policy
- OPENDJ-4920: LDAPS connections which are still inside handshake do not get idle closed
- OPENDJ-4474: Changing the JE db-logging-level to a non-allowed value disables the backend on restart
- OPENDJ-4881: Updates via REST2LDAP fail if record does not contain the necessary object class
- OPENDJ-4589: dsconfig --offline is not case-insensitive
- OPENDJ-4948: Certificate Mappers fail to use only local backends when matching user entries
- OPENDJ-4106: Incorrect error when importing bad LDIF on setup
- OPENDJ-4947: SASL DIGEST-MD5: bind request failed with protocol error
- OPENDJ-4935: Topology with three or more replication servers generates many outdated RS to RS update messages
- OPENDJ-5039: Upgrade task tries to move the opendmk-jarfile to a wrong path on instances with split instance/tool folders
- OPENDJ-4325: Changelog searches requesting changelogCookie are very slow
- OPENDJ-4714: SSL handshake now sends 16KB list of CA issuer DNs
- OPENDJ-4109: The ldappasswordmodify command fails when requested through a directory proxy server

- OPENDJ-4967: REST2LDAP UndeliverableException occurring when a referenced entity cannot be fetched
- OPENDJ-4625: Changelog range searches miss entries
- OPENDJ-5012: Replication: reset-change-number fails when DJ exposes different public naming contexts (replicated or not)
- OPENDJ-4851: Exception when uninstalling/stopping replication topology
- OPENDJ-4852: Backup with --backupAll misses a few backends
- OPENDJ-4226: Online list backups command throws error
- OPENDJ-4312: addrate raises NoSuchElementException when using numusers
- OPENDJ-4898: Server fails to ignore attempts to abandon certain operations
- OPENDJ-4693: Online rebuild-index command ends with a benign error message
- OPENDJ-4059: dsconfig --bindDN should default to "cn=Directory Manager"

Chapter 5
Documentation Updates

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

Documentation Change Log

Date	Description
2021-12-14	• 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. • Updated "Key Fixes" to include OPENDJ-4729.
2019-09-26	• Updated "Preventing Interference With Antivirus Software" in the *Installation Guide* to clarify how to prevent interference.
2018-11-26	Fixed paths to point to the correct `ads-truststore` file location, which is now `/path/to/opendj/db/ads-truststore` by default.
2018-06-27	Clarified that Solaris is no longer supported in "Removed Functionality".
2018-06-18	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")`.
2018-05-13	Added a step to "To Upgrade Replicated Servers" in the *Installation Guide* showing how to add missing privileges to the global administrator account. These privileges are required when using the **dsreplication status** command.
2018-05-04	In addition to the changes described in "*What's New*" and "*Compatibility*", the following important changes were made to the documentation: • A Deployment Guide has been added in this release. • "Initializing Replicas" in the *Administration Guide* has been updated to clarify the tradeoffs to consider when deciding how to initialize replication. The section has also been updated to demonstrate the **dsreplication initialize** command, which is the subcommand to use when initializing a single replica. • "Enforcing Strong Passwords and Strong Password Storage" in the *Security Guide* has been updated to improve the recommendations regarding password storage.

Date	Description
	• "Adding a REST to LDAP Mapping for a Custom Object" in the *Developer's Guide* has been added to demonstrate how to configure your REST to LDAP APIs. • "Adding Subentry Password Policies" in the *Developer's Guide* has been added to demonstrate how to use this feature with REST to LDAP. • The Javadoc now describes all ForgeRock classes and interfaces required to write server plugins and LDAP client applications.

Appendix A. Release Levels and Interface Stability

This appendix includes ForgeRock definitions for product release levels and interface stability.

Some interfaces are labelled as Evolving in the body of the documentation. In addition, the following rules apply:

- All Java APIs are Evolving, except `com.*` packages, which are Internal/Undocumented.
- The class `org.forgerock.opendj.ldap.CoreMessages` is Internal.
- The configuration, user, and application programming interfaces for RESTful access over HTTP to directory data are Evolving. This includes interfaces exposed for the HTTP connection handler, its access log, and also the REST to LDAP gateway.
- Text in log messages should be considered Internal. Log message IDs are Evolving.
- The default content of `cn=schema` (LDAP schema) is Evolving.
- Newly Deprecated and Removed interfaces are identified in "*Compatibility*".
- Interfaces that are not described in released product documentation should be considered Internal/Undocumented. For example, the LDIF representation of the server configuration, `config.ldif`, should be considered Internal.

A.1. ForgeRock Product Release Levels

ForgeRock defines Major, Minor, Maintenance, and Patch product release levels. The release level is reflected in the version number. The release level tells you what sort of compatibility changes to expect.

Release Level Definitions

Release Label	Version Numbers	Characteristics
Major	Version: x[.0.0] (trailing 0s are optional)	• Bring major new features, minor features, and bug fixes • Can include changes even to Stable interfaces • Can remove previously Deprecated functionality, and in rare cases remove Evolving functionality that has not been explicitly Deprecated • Include changes present in previous Minor and Maintenance releases
Minor	Version: x.y[.0] (trailing 0s are optional)	• Bring minor features, and bug fixes • Can include backwards-compatible changes to Stable interfaces in the same Major release, and incompatible changes to Evolving interfaces • Can remove previously Deprecated functionality • Include changes present in previous Minor and Maintenance releases
Maintenance, Patch	Version: x.y.z[.p] The optional `.p` reflects a Patch version.	• Bring bug fixes • Are intended to be fully compatible with previous versions from the same Minor release

A.2. ForgeRock Product Interface Stability

ForgeRock products support many protocols, APIs, GUIs, and command-line interfaces. Some of these interfaces are standard and very stable. Others offer new functionality that is continuing to evolve.

ForgeRock acknowledges that you invest in these interfaces, and therefore must know when and how ForgeRock expects them to change. For that reason, ForgeRock defines interface stability labels and uses these definitions in ForgeRock products.

Interface Stability Definitions

Stability Label	Definition
Stable	This documented interface is expected to undergo backwards-compatible changes only for major releases. Changes may be announced at least one minor release before they take effect.
Evolving	This documented interface is continuing to evolve and so is expected to change, potentially in backwards-incompatible ways even in a minor release. Changes are documented at the time of product release. While new protocols and APIs are still in the process of standardization, they are Evolving. This applies for example to recent Internet-Draft implementations, and also to newly developed functionality.
Deprecated	This interface is deprecated and likely to be removed in a future release. For previously stable interfaces, the change was likely announced in a previous release. Deprecated interfaces will be removed from ForgeRock products.
Removed	This interface was deprecated in a previous release and has now been removed from the product.
Technology Preview	Technology previews provide access to new features that are evolving new technology that are not yet supported. Technology preview features may be functionally incomplete and the function as implemented is subject to change without notice. DO NOT DEPLOY A TECHNOLOGY PREVIEW INTO A PRODUCTION ENVIRONMENT. Customers are encouraged to test drive the technology preview features in a non-production environment and are welcome to make comments and suggestions about the features in the associated forums. ForgeRock does not guarantee that a technology preview feature will be present in future releases, the final complete version of the feature is liable to change between preview and the final version. Once a technology preview moves into the completed version, said feature will become part of the ForgeRock platform. Technology previews are provided on an “AS-IS” basis for evaluation purposes only and ForgeRock accepts no liability or obligations for the use thereof.
Internal/Undocumented	Internal and undocumented interfaces can change without notice. If you depend on one of these interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Appendix B. Getting Support

For more information and resources about DS and ForgeRock support, see the following sections:

B.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.

B.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.

B.3. 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.