Release Notes

/ Directory Services 7

Latest update: 7.0.2

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

Overview

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.

Quick Start

 What's New Discover new features and improvements in this version.	 Prepare for Deployment Learn about the requirements for running DS software in production.	 Check Compatibility Review key implementation changes and compatibility with previous deployments.
 Review Fixes Review bug fixes, limitations, and open issues.	 Check Doc Updates Track important changes to the documentation.	 Get Support Find out where to get professional support and training.

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.

Chapter 1
What's New

Maintenance Releases

ForgeRock maintenance releases contain a collection of fixes and minor RFEs that have been grouped together and released as part of our commitment to support our customers. ForgeRock maintenance releases also bundle the latest maintenance update of DS software dependencies, which may include fixes and minor RFEs.

For general information on ForgeRock's maintenance and patch releases, see Maintenance and Patch Availability Policy.

New in 7.0.2

There are no new features in DS 7.0.2, only bug fixes.

DS 7.0.2 is the latest release targeted for DS 7.0.0, and DS 7.0.1 deployments. It can be downloaded from the *ForgeRock Backstage* website.

To view the list of fixes in this release, see "Fixed in 7.0.2".

The release can be deployed as an initial deployment or updated from an existing DS 7.0.x deployment.

New in 7.0.1

- The DS password synchronization plugin for IDM now supports OAuth 2.0 access token bearer authentication.

  For details, see *Synchronizing Passwords With ForgeRock Directory Services (DS)* in the IDM *Password Synchronization Plugin Guide*.

- DS command options that have secrets as arguments now support `:env` and `:file` modifier suffixes. Use these with the following options to provide the secret in an environment variable (`:env`), or in a file (`:file`):

  - `--bindPassword[:env|:file]`
  - `--deploymentKeyPassword[:env|:file]`

- `--keyStorePassword[:env|:file]`
- `--monitorUserPassword[:env|:file]`
- `--rootUserPassword[:env|:file]`
- `--set[:env|:file]` (for setup profile parameters)
- `--trustStorePassword[:env|:file]`

For example, if the bind password is stored in a `~/.pass` file, use `--bindPassword:file ~/.pass`. If the password is stored in the environment variable `PASS`, use `--bindPassword:env PASS`.

- The **supportextract** command now uses the **jcmd** command, if available, for heap dumps. Otherwise, it uses the **jmap** command.

  Issue: OPENDJ-7662

New in 7.0.0

Access Control	+ *Aliases for controls and extended operations* This release supports use of aliases in addition to OIDs for LDAP controls and extended operations in ACIs, making those ACIs significantly more human-readable. For details, see "Directory Server ACIs" in the *Security Guide*. Since previous releases support only OIDs, only use aliases in ACIs after upgrading all directory servers. Otherwise, older servers will log warning messages for the unrecognized aliases, such as the following: `Access Control Instruction (ACI) targetcontrol expression value "value" is invalid. A valid targetcontrol keyword expression value requires one or more valid control OID strings in the following format: oid [|| oid1] ... [|| oidN]`
Authentication	+ *Multiple identity mappers* The following configuration objects can now reference multiple identity mappers: • "CRAM-MD5 SASL Mechanism Handler" • "DIGEST-MD5 SASL Mechanism Handler" • "GSSAPI SASL Mechanism Handler"

- "HTTP Basic Authorization Mechanism"
- "HTTP OAuth2 Authorization Mechanism"
- "Password Modify Extended Operation Handler"
- "Plain SASL Mechanism Handler"
- Global configuration, proxied-authorization-identity-mapper

When resolving the identity, the server uses the first identity mapper that finds a match. If multiple identity mappers match different entries, however, then the server returns LDAP error code 19, Constraint Violation.

For background information, see "Identity Mappers" in the *LDAP User Guide*.

Backup and Restore

+ *New, simplified implementation with cloud storage support*

The release provides a new, simplified implementation for DS backup and restore operations:

- The new implementation replaces backup archives with collections of backup files.

  The collection includes backend files and backup metadata. The files always follow the same layout, regardless of what you back up.

  You manage backup files by retaining an entire backup directory. You are no longer required to use a separate backup directory for each backend.

- You can now stream backup files directly to cloud storage, and restore directly from cloud storage.

- You no longer have to make a choice between full and incremental backup operations. Backup operations are incremental by design. When you reuse the same backup directory, the process only backs up new data.

- The new implementation includes a **purge** command for removing old backup files. You can purge old files either as an external command, or as a server task.

- In the event of a disaster, you can restore from a backup directory stored off-site using only the deployment key and password, and a backup copy of the server configurations.

  The new implementation protects (encrypts) the backup encryption keys with the shared master key. It stores the encrypted encryption keys in the backup files.

  You no longer need to configure replication between new replicas and a server from the existing topology. Instead, you first set up replacement replicas with the deployment key and password, restoring the backed up

server configurations to match those servers lost in the disaster. You then restore the data using the off-site backup directory.

- The new implementation always signs and verifies the integrity of backups, and always encrypts backup files.

  The new implementation encrypts the keys used for signing and encryption with the shared master key. It stores the encrypted keys in the backup files.

  You can verify the integrity and ability to decrypt backups before restoring a backend.

- The new implementation makes it possible to list and verify backups while the server is online.

- The new implementation improves restore performance compared to restore of incremental backups in previous versions.

  The previous implementation restored files from the full backup archive, and then restored files from each incremental backup archive. Files could be restored and then removed, or restored multiple times.

  The new implementation only restores one version of each file in the backup directory.

- A new command, **dsbackup**, replaces the **backup** and **restore** commands.

  The **dsbackup** command performs operations formerly performed using separate commands:

  - **dsbackup create** performs backup operations.
  - **dsbackup list** displays a summary of available backups, and lets you verify them.
  - **dsbackup purge** removes old backup files.
  - **dsbackup restore** performs restore operations.

- The new **dsbackup restore** command has a `--backendName` option, which lets you restore only the specified backend.

For examples, see "*Backup and Restore*" in the *Maintenance Guide*.

Cloud Deployments

+ *Ready for Docker and Kubernetes*

This release lifts restrictions on running DS servers in Docker and Kubernetes deployments. Many individual improvements make this possible, including the following:

- Replication improvements let you scale the number of DS replicas in your stateful sets up and down.

	• The new **dsrepl** command runs well in Docker containers. ForgeRock supports customers deploying DS in Docker containers and Kubernetes platforms, as highlighted in the important note under "*Requirements*". To get started, try the following: • Use the `forgeops` repository and the unsupported, evaluation-only base images for the ForgeRock Identity Platform. The images are available in ForgeRock's public Docker registry. For details, see *Base Docker Images* in the ForgeRock DevOps documentation. • Build your own sample DS Docker image. Unpack the .zip distribution, then see the `opendj/samples/docker/README.md` file for instructions.
Collective Attributes	+ *Relative parent support* DS servers now support specifying the relative parent in collective attribute subentries. For details, see "Inherit From a Parent Entry" in the *Configuration Guide*.
Data Storage	+ *Shared database cache by default* By default, DS servers now share cache memory among JE database backends. The server keeps JE database internal and leaf nodes in the database heap cache. For existing servers, the **upgrade** command does not change the database cache behavior. Consider setting the global property `je-backend-shared-cache-enabled:true`, and the JE backends' properties `db-cache-mode:cache-ln` after upgrade. For details, read the following documentation: • "Database Cache Settings" in the *Maintenance Guide* • "Java Settings" in the *Maintenance Guide* • je-backend-shared-cache-enabled in the *Configuration Reference* • db-cache-mode in the *Configuration Reference* + *Newer JE*

	This release upgrades JE backend databases to Berkeley DB Java Edition 18.3.12. **Important** Different DS server versions continue to replicate data during the upgrade process. However, the JE upgrade has the following implications for the portability of local DS data. Once you upgrade the data in a JE backend database: • You cannot downgrade a directory server without also restoring JE backend data from a pre-upgrade server. • You cannot restore backups of an upgraded JE backend on a pre-upgrade directory server. In addition, several JE backend properties that affect cache sizing and database maintenance can now be changed at runtime without restarting the backend. For details, see "JE Backend".
Data Encryption	+ *Portable encrypted data* DS servers now store symmetric keys, encrypted with the shared master key, with the data they encrypt. It is no longer necessary for disaster recovery to maintain a file system backup of a server from each replication topology in your deployment. It is now sufficient to keep the backup directory and a means to recover the shared master key. As long as a server has the same shared master key as the server that encrypted the data, it can recover symmetric keys needed to decrypt data. Be aware that this feature is new, and not provided in previous versions of DS software. Replication is fully compatible with previous server versions, but backup files are not. *For this feature to work, you must use a backup from an upgraded or new server.* + *GCM with AES* DS directory servers now support Galois/Counter Mode (GCM) with AES for encrypted data confidentiality. GCM is efficient and improves integrity protection for encrypted backend data.

	Set the data encryption cipher transformation, as described in "*Data Encryption*" in the *Security Guide*. The default setting for the backend property, cipher-transformation, is now `AES/GCM/NoPadding`.
Email Notifications	+ *Secure, authenticated connections* Email notifications now support SMTP authentication and use of TLS. For details, see "Send Account Status Mail" in the *Maintenance Guide*, and "Mail Server".
Interoperability	+ *Microsoft AD range retrieval support* DS software now supports the * character in malformed attribute options for interoperability with the Microsoft Active Directory "range retrieval" mechanism.
Logging	+ *Field whitelisting* ForgeRock Common Audit loggers now whitelist all fields that are safe to log by default. The whitelist is processed before the blacklist, so blacklist settings overwrite the whitelist defaults. For details, see "Whitelist Log Message Fields" in the *Logging Guide*. + *Error messages to standard output* DS servers can now send error messages to standard output. For details, see "Log Errors to Standard Output" in the *Logging Guide*. + *More information about operations in access logs* DS servers now record additional information about LDAP operations in access log messages: • For LDAP bind operations, the security strength factor (SSF) negotiated for secure client connections appears in the `response` field of the access log message. For example: `{..."request":{"protocol":"LDAPS","operation":"BIND"...}..."response":{..."additionalItems":"ssf=128"},...}`

	• For persistent searches, the log messages include `"additionalItems":"persistent"`. + *Details when a connection handler fails to start* When a connection handler fails to start, DS servers now log an error message indicating the cause.
Monitoring	+ *Monitor account replicated by default* DS servers now replicate the monitor user created at setup time (default DN: `uid=monitor`). This lets commands like **dsrepl status** use the same account credentials to retrieve monitoring information from all servers. You can use the account in the same way for multi-server monitoring operations.
Networking	+ *Advertised listen address* DS servers now have a new, required, global property, advertised-listen-address. This setting specifies the hostname or IP address that clients should use for connecting to the server. The `advertised-listen-address` can be multi-valued in systems with multiple network interfaces. DS servers also now have a global property, listen-address. The `listen-address` property can be set to the wildcard IP address, `0.0.0.0`, but the `advertised-listen-address` property cannot. By default, replication and connection handlers inherit their settings for listen addresses from these global properties. This improvement lets DS servers make fewer DNS requests than before. When setting up a new server, the **setup** command sets the `advertised-listen-address` property to the IP address or the FQDN provided as the `--hostname` argument. During upgrade, the value for the `advertised-listen-address` property is assigned using the hostname derived from administrative data under `cn=admin data`. If any `listen-address` properties are set to the same value, then those settings are removed during upgrade, and the values are inherited instead.
Passwords	+ *SCRAM SASL support* DS software now supports Salted Challenge Response Authentication Mechanism (SCRAM) SASL binds.

A SASL SCRAM mechanism provides a secure alternative to transmitting plaintext passwords during binds. It is an appropriate replacement for DIGEST-MD5 and CRAM-MD5.

With a SCRAM SASL bind, the client must demonstrate proof that it has the original plaintext password. During the SASL bind, the client must perform computationally intensive processing to prove that it has the plaintext password. This computation is like what the server performs for PBKDF2, but the password is not communicated during the bind.

Once the server has stored the password, the client pays the computational cost to perform the bind. The server only pays a high computational cost when the password is updated, for example, when an entry with a password is added or during a password modify operation. A SASL SCRAM mechanism therefore offers a way to offload the high computational cost of secure password storage to client applications during authentication.

Passwords storage using a SCRAM storage scheme is compatible with simple binds and SASL PLAIN binds. When a password is stored using a SCRAM storage scheme, the server pays the computational cost to perform the bind during a simple bind or SASL PLAIN bind.

The SCRAM password storage scheme must match the SASL SCRAM mechanism used for authentication. In other words, SASL SCRAM-SHA-256 requires a SCRAM-SHA-256 password storage scheme. SASL SCRAM-SHA-512 requires a SCRAM-SHA-512 password storage scheme.

DS software offers the following in the configuration for new servers:

Password Storage Scheme	SASL Mechanism
SCRAM-SHA-256	SCRAM-SHA-256
SCRAM-SHA-512	SCRAM-SHA-512

For additional information, see "Password Storage" in the *Security Guide* for the server, and "Gateway LDAP Connections" in the *HTTP User Guide* for the REST to LDAP gateway.

+ *Full-featured, replicated password policies*

DS servers now support LDAP subentry password policies that match all features available in per-server password policies.

Servers store subentry policies in the directory data, and therefore replicate them. This improvement significantly simplifies password policy management across multiple replicas.

For details, see "DS Subentry Password Policies" in the *Security Guide*. Many samples in the documentation now demonstrate features of the improved subentry password policies.

+ *Stronger password storage schemes*

DS servers now support additional password storage schemes, `PBKDF2-HMAC-SHA256` and `PBKDF2-HMAC-SHA512`.

The new password storage schemes use SHA-256 and SHA-512 hash-based message authentication code settings. The `PBKDF2` password storage scheme uses SHA-1.

To migrate passwords to a new storage scheme, see "Deprecate a Password Storage Scheme" in the *Security Guide*.

+ *128-bit salt*

Salted hashed password storage schemes now use 128-bit salt when generating a hash.

This change applies to the following password storage schemes:

PBKDF2
PBKDF2-HMAC-SHA256
PBKDF2-HMAC-SHA512
Salted MD5
Salted SHA-1
Salted SHA-256
Salted SHA-384
Salted SHA-512

+ *Rehash passwords*

You can now configure BCrypt and PBKDF2-based password storage schemes to recalculate password hashes after the iterations settings are changed. DS servers recalculate and store an account's password hash when the user binds successfully with their password.

For details regarding BCrypt, see the reference for the property rehash-policy. For details regarding PBKDF2-based schemes, see the reference for the property rehash-policy.

+ *Password quality advice*

DS servers support a new control to request password quality advice when changing a password. Should the request fail due to low password quality, the response control indicates which password validator settings led to the failure.

	The **ldappasswordmodify** and **ldapmodify** commands support the new control. Use them to test and debug password policy validation settings. The new LDAP control has interface stability: Evolving. It may be removed in a future release, or replaced with a more general mechanism. For details, see "Check Password Quality" in the *LDAP User Guide*, and "Check Password Quality" in the *HTTP User Guide*.
Performance	+ *Faster export* The **export-ldif** command can now complete an export up to twice as fast as before. This improvement is particularly useful with large data sets including tens or hundreds of millions of entries. + *Faster REST to LDAP performance* DS servers now perform better for REST to LDAP searches, and operations that rely on ETags for MVCC.
Proxy	+ *Mutual TLS with LDAP servers* The **setup** command now lets a proxy backend bind to remote servers with mutual TLS. The setup profile for a proxy server configures the server to use mutual TLS to authenticate when binding to backend servers. As a result, you must provision the key manager for the proxy with the proxy service account keys, and include the certificate in the proxy user account when using the DS proxy server setup profile. For details, see "*Install Directory Proxy*" in the *Installation Guide*. + *More Information* When setting up new DS replicas, use the `ds-proxied-server` setup profile to prepare the replicas for use with new DS proxy servers. For details, see "*Install DS For Use With DS Proxy*" in the *Installation Guide*.
REST	+ *Path references* REST to LDAP mappings now support references by resource paths, simplifying access to all resource fields. RESTful clients can use this to issue graph-like queries. For example, the following path and query filter returns the groups that Babs Jensen's manager belongs to: `/users/bjensen?_fields=/manager/group`

For an example, see "Graph-Like Queries" in the *HTTP User Guide*.

To demonstrate this feature, the sample REST to LDAP mapping now uses resource paths. The configuration is simpler than the configuration with base DN references.

For example, this excerpt shows a manager reference from the version that uses a base DN:

```
{
  "manager": {
    "type": "reference",
    "ldapAttribute": "manager",
    "baseDn": "..",
    "primaryKey": "uid",
    "mapper": {
      "type": "object",
      "properties": {
        "_id": {
          "type": "simple",
          "ldapAttribute": "uid",
          "isRequired": true
        },
        "displayName": {
          "type": "simple",
          "ldapAttribute": "cn",
          "writability": "readOnlyDiscardWrites"
        }
      }
    }
  }
}
```

The same manager reference using a resource path now looks like this:

```
{
  "manager": {
    "type": "reference",
    "resourcePath": ".."
  }
}
```

The latter definition ensures access to all fields defined for the referenced resource.

+ *Reverse references*

REST to LDAP mappings now support reverse references.

Reverse references are similar to the `isMemberOf` LDAP attribute used for groups. For example, use a reference reference mapping to lists a user's devices, or to list a manager's reports:

```
{
  "reports": {
    "type": "reverseReference",
    "resourcePath": "..",
    "propertyName": "manager"
  }
}
```

For an example in context, see "Reverse References" in the *HTTP User Guide*.

+ *Password quality advice*

REST to LDAP now supports `passwordQualityAdvice` and `dryRun` query string parameters.

The `passwordQualityAdvice` parameter relies on the DS LDAP password quality advice control, OID `1.3.6.1.4.1.36733.2.1.5.5`, which users must have access to request. The `dryRun` parameter relies on the LDAP no-op control, OID `1.3.6.1.4.1.4203.1.10.2`.

The password quality advice control and the `passwordQualityAdvice` parameter have interface stability: Evolving. They may be removed in a future release, or replaced with a more general mechanism.

For details, see "Check Password Quality" in the *HTTP User Guide*.

+ *Account usability support*

REST to LDAP now includes an `accountUsability` action.

For details, see "Account Usability Action" in the *HTTP User Guide*.

+ *SASL EXTERNAL and SASL SCRAM support*

The REST to LDAP gateway now supports SASL EXTERNAL and SASL SCRAM binds.

For details, see "Gateway LDAP Connections" in the *HTTP User Guide*.

+ *Per-Server password policies over REST*

DS servers now let you create per-server (configuration-based) password policies over REST.

For an example, see "Per-Server Password Policies" in the *HTTP User Guide*.

+ *Optional JSON for references*

	The REST to LDAP gateway supports using attributes with `NameAndOptionalJSON` syntax as references. For details, see "API Configuration" in the *HTTP User Guide*.
Replication	+ *Replication at setup time* The **setup** command now lets you configure replication at setup time. You therefore no longer need to get all peer servers running before configuring replication. The server begins replicating with peer servers when it comes online, and when it can contact the peers. For this reason, the **setup** command no longer starts the server by default. To ensure replication proceeds smoothly from the beginning, *finish configuring the server before starting it for the first time*. These new **setup** command options enable replication: • When you set the `-r, --replicationPort` option, the server runs a replication service and maintains a changelog. If you add local application data at setup time, the server replicates the data with other replicas. There is no need to configure and initialize replication separately. • When you set the `--bootstrapReplicationServer` option, the server contacts the specified replication server(s) to discover peer replicas and replication servers. This option is required when replicating between multiple servers. Use this option multiple times to specify redundant bootstrap servers for availability. Specify the same list of bootstrap servers each time you set up a replica. Your first bootstrap server(s) must have replication ports, because the first bootstrap server(s) must play the replication server role. For examples, see the Installation Guide. + *New command to manage replication* After configuring servers to replicate as part of the setup process, use the new **dsrepl** command to manage replication. For details, see "*Replication*" in the *Configuration Guide*. + *String-based server IDs*

This release lets you set server IDs to alphanumeric strings, such as `ds1-us-west`.

When you set a server ID, take care to choose a relatively short string.

The server ID appears in historical data values that include a change sequence number in the *Getting Started*. For example, it shows up in monitoring metrics, and in the values of `ds-sync-state` and `ds-sync-hist` attributes in application data on DS replicas. As a result, historical data is potentially easier to interpret, but larger than in previous versions where server IDs were numbers.

+ *One server ID per server*

Servers are now identified by a single, global server ID. For details, see server-id.

For new servers, use the **setup** command to specify the server ID, or accept the generated default string.

For existing servers, the **upgrade** command derives the ID in the following way:

1. The command the existing global server ID, if available.
2. Otherwise, the command uses the first server ID found in `cn=admin data`. Other server ID values are no longer used.
3. If replication has not yet been configured, the commmand generates a new ID for the server.

+ *One group ID per server*

Servers now have a single, global group ID. For details, see group-id.

For existing servers with group IDs, the **upgrade** command determines which ID is used most, and uses that ID as the single, global ID.

+ *Replication delay metrics*

This release introduces replication receive delay and replay delay monitoring metrics. These metrics provide the best means yet to help you estimate whether the data in your directory server replicas is converging toward a consistent state.

For details, see "Replication Delay (LDAP)", or "Replication Delay (Prometheus)" in the *Monitoring Guide*.

+ *Replay performance*

This release improves replication replay performance, reduces disk space used by the replication changelog database, and reduces replication delay in deployments under extreme load.

+ *Replication of offline LDIF changes*

Servers now replicate changes made offline to an LDIF backend. The server replicates the offline changes once it starts again.

+ *Automatic purge of stale replicas*

This release purges out-of-date replicas from the changelog. The replica is purged when it has been out of contact for longer than the replication purge delay.

This enables DS servers to eventually discard information about replicas that you have removed from service, for example.

You can also use the **dsrepl purge-meta-data** to eliminate stale historical data. For details, see "Manual Purge" in the *Configuration Guide*.

+ *Exclude domains from changelog indexes*

This release introduces a new replication server property to exclude domains from the changelog indexes, changelog-enabled-excluded-domains. Use this to prevent applications that read the external change log from having to process update notifications for entries that are not relevant to them.

This property eliminates the need for a separate external changelog domain configuration.

For an example, see "Exclude a Domain" in the *Configuration Guide*.

+ *CTS excluded from changelog indexing*

The `am-cts` setup profile now excludes the CTS base DN from change number indexing.

	There is no need to update the changelog configuration manually after installing a new DS replica for as a CTS store.
	+ *More details about unresolved conflicts*
	DS servers now log additional information about naming conflicts, which helps you identify the server that generated the conflicting operation.
Samples	+ *Sample for building custom Docker images*
	The DS server distribution now includes a sample `Dockerfile` and related files for building custom DS Docker images.
	+ *Updated sample for Grafana and Prometheus*
	The DS server distribution now includes an updated sample monitoring dashboard for use with Grafana and Prometheus.
Schema	+ *Name and optional JSON*
	DS servers now support an attribute syntax for a DN optionally prepended with a JSON object. The associated matching rules let the server index and match the prepended JSON, or ignore it. For details, see: • "Name and Optional JSON" in the *LDAP Schema Reference* • "nameAndOptionalJsonEqualityMatchingRule" in the *LDAP Schema Reference* • "nameAndOptionalCaseExactJsonIdEqualityMatch" in the *LDAP Schema Reference* • "nameAndOptionalCaseIgnoreJsonIdEqualityMatch" in the *LDAP Schema Reference* • "jsonFirstComponentCaseExactJsonQueryMatch" in the *LDAP Schema Reference* • "jsonFirstComponentCaseIgnoreJsonQueryMatch" in the *LDAP Schema Reference*
	+ *Require TRUE or FALSE boolean values*

DS servers now support an option to require strict compliance for boolean attribute values.

By default, DS servers accept a range of values for boolean attributes. For details, see strict-format-boolean.

Security

+ *Secure by default*

Default settings for new DS servers are more secure than before.

The explicit `--productionMode` option has been removed, as server configurations and profiles are now secure by default. New server installations require:

Secure connections

All operations except bind requests and StartTLS requests, and base object searches on the root DSE, require secure connections.

This behavior is governed by the global configuration property, unauthenticated-requests-policy, which is now set to `allow-discovery`, instead of `allow`, unless the last setup profile applied is the `ds-evaluation` profile.

For details on securing connections, see "*Secure Connections*" in the *Security Guide*.

Authentication

By default, servers deny anonymous access to most LDAP operations, controls, and extended operations.

For details on access control, see "*Access Control*" in the *Security Guide*.

Additional access policies

By default, servers deny access to directory data. You must configure access policies to grant access to directory data. For details on granting access, see "*Access Control*" in the *Security Guide*.

Only the evaluation setup profile is more lenient. It grants global permission to perform operations over insecure connections, and open access to sample Example.com data. For details, see "Learn About the Evaluation Setup Profile" in the *Installation Guide*.

Stronger passwords

Passwords must have at least 8 characters. Common passwords are rejected.

For details on changing password policy, see "Configure Password Policies" in the *Security Guide*.

Permission to read log files

Log files are now read/write only by the DS server user.

For details on log file permissions, see "File Permissions" in the *Security Guide*.

As the upgrade process preserves the existing configuration, upgraded servers are not affected.

Review the changes in "Default Security Settings".

+ *Simple private PKI*

The **setup** and **dskeymgr** commands simplify creation and management of a public key infrastructure (PKI).

This release introduces the concept of a deployment key and deployment key password. The deployment key and password serve as an alternative to a private CA, simplifying evaluation, development, and testing, and managing directory services. They also serve to derive a shared master key to protect secret keys. The deployment key and password are required as part of the setup process. For details, see "Key Management" in the *Security Guide*.

When you use an existing CA, you can continue to use key pairs with CA-signed certificates.

For public-facing directory services, you can continue to configure connection handlers with additional key and trust manager providers using certificates signed by a well-known CA. For details, see "Key Management" in the *Security Guide*.

To manage deployment keys, key pairs, CA certificates, and master keys after setting up a server, use the **dskeymgr** command.

Many examples in the documentation now demonstrate use of deployment keys and passwords.

+ *Keystores reload without restart*

DS servers now reload file-based keystores and truststores when their contents change.

This lets you rotate certificates and keys without restarting the key manager or trust manager components.

+ *Simple key rotation*

This release greatly simplifies rotating the key pairs used to secure replication connections. By default, replication now uses the same keys as the other connection handlers.

For details on changing key pairs, see "Key Management" in the *Security Guide*.

+ *Alternative PKCS#11 types*

PKCS#11 key managers and trust managers now let you set the keystore or truststore type. The default type is `PKCS11`.

If your JVM supports other types, set the keystore or truststore type with one of the following properties:

- key-store-type
- trust-store-type

+ *Multiple trust managers*

The following configuration objects can now reference multiple "Trust Manager Provider" objects:

- "Administration Connector"
- "HTTP Connection Handler"
- "LDAP Connection Handler"

Use this feature to allow trust for both well-known CAs whose certificates are stored in the JVM truststore, and internal or deployment-specific CAs whose certificates are stored in a separate truststore.

+ *Multiple certificate mappers*

An external SASL mechanism handler can now reference multiple certificate-mapper configurations. The server uses the first certificate mapper that finds a successful match.

+ *Indexes for certificate attributes*

When you create a user data backend using the `ds-user-data` setup profile, the setup process now configures equality indexes for the `ds-certificate-`

`fingerprint` and `ds-certificate-subject-dn` attributes. Certificate mappers use these indexes during certificate-based authentication.

+ *Richer access log messages*

DS servers now record additional items in access log messages when multiple password policy subentries apply to a user. The messages are logged only for bind, add, and modify operations. The messages show the DN of the user having more than one applicable policy, and the DN of the policy the server actually used for the operation. The server logs a message such as the following for a bind request with two conflicting policies:

```
"additionalItems":{"pwdpolicywarning":"Found 2 conflicting password
 policy subentries for user <user-dn>,
used <policy-dn>","ssf":"0"}
```

As described in "Assign Password Policies" in the *Security Guide*, you must not assign more than one password policy to the same account.

Tools

+ *New setup-profile command*

A new command, **setup-profile**, enables configuration of setup profiles following initial installation. Use the **setup-profile** command when the server is offline.

This command is intended for use in DevOps deployments where you apply additional configuration to a base image that is the same for all deployments.

If you have changes that apply to each server you set up, you can create and maintain your own setup profile. For details, see "Create Your Own" in the *Installation Guide*.

+ *Configurable domains and base DNs*

All **setup** command profiles, except the `ds-evaluation` profile, now allow you to set the domain or the base DN. For details, see "*Setup Profiles*" in the *Installation Guide*.

+ *Generate systemd service files*

The **create-rc-script** command now produces a `systemd` service file when you use the `--systemdService` option.

+ *Generate user entries for evaluation*

The `ds-evaluation` setup profile now lets you generate an arbitrarily large number of similar user entries. By default, the profile adds 100,000 generated

users in addition to users previously included, such as Babs Jensen and Kirsten Vaughan.

Each user entry has a `uid` RDN like `user.number`. Each user entry's password is `password`.

The capability replaces the **setup** command option `-d, --sampleData`.

+ *Proxied authorization for rate tools*

The **addrate**, **modrate**, and **searchrate** commands now support proxied authorization with the `-Y, --proxyAs {authzID}` option.

+ *Support for formatted integers*

Formatted integers can now be supplied to some integer arguments, making commands easier to read.

When setting the number of generated sample entries as an argument to the **setup** command, and when setting integer arguments for the **addrate**, **authrate**, **modrate**, and **searchrate** commands, you can now use formatted integers. For example, the following are equivalent to `10000000`:

```
10,000,000
10.000.000
"10'000'000"
10_000_000
"10 000 000"
```

Templates for the **makeldif** command can also accept formatted integers for numbers declared in a subordinate template.

+ *Task management*

The **manage-tasks** command now has `--status` and `--type` options.

When used with the `--summary` option, these options filter the list to include only tasks of the specified type and status. The option arguments are case insensitive, and must be provided in the JVM locale. For example, to list only unscheduled tasks on a JVM with the French locale, use `--status "non planifié"` instead of `--status unscheduled`.

+ *Set task IDs and descriptions*

When you schedule a task, you can now set its identifier with the `--taskId` option, and its description with the `--description` option. The identifiers and descriptions appear in output and messages that describe the task.

These new options are especially useful for recurring tasks. Use the task identifier when managing the task in subsequent commands, for example.

+ *No changes when reading JE backends offline*

The following tools now never write to JE backend databases when reading JE information:

- **backendstat**
- **export-ldif**
- **verify-index**

+ *Status output improvements*

The **status** command now displays the same types of information independently of the server configuration, and regardless of whether the command runs in online or offline mode.

The command still displays more detailed information in online mode than in offline mode.

+ *Supportextract improvements*

- The **supportextract** command now also collects:
  - The directory superuser and monitor user account files.
  - The archived configuration files.
  - The profile and backend database version files.
  - Information about the changelog database.
  - The `server.out` log file before capturing stack traces.
  - The server PID in a message in the tool's log.
  - The `cpuinfo`, `meminfo`, `slabinfo`, and `buddyinfo` files on UNIX and Linux systems.
  - Stack traces with **jcmd** tool, falling back to the **jstack** tool, and then to **sigquit** (or **kill -3** on Linux) as necessary.
  - Environment variables used in configuration expressions.

- The extract generated by the tool is now compatible with the Java 11 JVM unified logging framework.

+ *Byte-by-byte LDIF comparisons*

The **ldifdiff** command now supports a new `-x, --exactMatch` option for byte-by-byte LDIF comparisons.

This is useful for comparing LDAP schema files, for example.

Chapter 2
Requirements

Important

ForgeRock supports customers deploying DS in Docker containers and Kubernetes platforms, as well as bare metal and VM deployments, provided you follow the hardware and software requirements specified here.

Other versions and alternative environments might work as well. When opening a support ticket for an issue, however, make sure you can also reproduce the problem on a combination covered here.

Downloads

The ForgeRock BackStage download 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.

File	Description
`DS-7.0.0.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. 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-7.0.2.msi`	Microsoft Windows native installer for the server software. By default, this installs files into a `C:\Program Files (x86)\OpenDJ\` directory.
`DS_7.0.2-1_all.deb`	Server software native packages for Debian and related Linux distributions.

File	Description
	By default, this installs files into an `/opt/opendj/` directory.
`DS-7.0.2-1.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-7.0.2.war`	Cross-platform DSML gateway web archive.
`DS-rest2ldap-servlet-7.0.2.war`	Cross-platform REST to LDAP gateway web archive.

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.

Memory

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.

As a rule of thumb, the RAM available for the server should be at least 1.5 to 2 times the total size of the database files on disk, or at minimum 2 GB.

Provide 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.

Disk Space

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 "Plan to Scale" in the *Deployment Guide*.

CPU Architectures

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.

Network

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.

Storage

Warning

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.

Operating Systems

Directory Services 7 software is supported on the following operating systems:

Supported Host Operating Systems

Operating System	Versions
Red Hat Enterprise Linux, Centos	7, 8
Amazon Linux	Amazon Linux 2018.03
SuSE	12, 15
Ubuntu	16.04 LTS 18.04 LTS
Windows Server	2016, 2019

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 set the journaling mode for data, see the options for your file system in the **mount** command manual page.

Maximum Open Files

DS servers must open many file descriptors when handling thousands of client connections.

Linux systems often set a limit of 1024 per user. That setting is too low to handle to handle thousands of client connections.

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, 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. Check the Linux system overall maximum as follows:

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

Maximum Watched Files

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, update the `/etc/sysctl.conf` file to change the setting permanently, and use the **sysctl -p** command to reload the settings:

```
$ echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf
[sudo] password for admin:
$ sudo sysctl -p
fs.inotify.max_user_watches = 524288
```

Antivirus Interference

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 "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.

Java

Directory Services software supports the following Java environments:

Supported Java Versions

Vendor	Versions
OpenJDK, including OpenJDK-based distributions: • AdoptOpenJDK/Eclipse Adoptium • Amazon Corretto • Azul Zulu • Red Hat OpenJDK ForgeRock tests most extensively with AdoptOpenJDK/Eclipse Adoptium.	11
Oracle Java	11

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.

Application Containers

The REST to LDAP and DSML gateway applications support the following web application containers:

Supported Web Application Containers

Container	Versions
Apache Tomcat	8.5, 9
IBM WebSphere Liberty	20.0.0.1
JBoss Enterprise Application Platform	7.2
Wildfly	12, 19

Third-Party Software

ForgeRock provides support for using the following third-party software when logging ForgeRock Common Audit events:

Software	Version
Java Message Service (JMS)	2.0 API
MySQL JDBC Driver Connector/J	8 (at least 8.0.19)
Splunk	8.0 (at least 8.0.2)

Tip

Elasticsearch and Splunk have native or third-party tools to collect, transform, and route logs. Examples include Logstash and Fluentd.

ForgeRock recommends that you consider these alternatives. These tools have advanced, specialized features focused on getting log data into the target system. They decouple the solution from the ForgeRock Identity Platform systems and version, and provide inherent persistence and reliability. You can configure the tools to avoid losing audit messages if a ForgeRock Identity Platform service goes offline, or delivery issues occur.

These tools can work with ForgeRock Common Audit logging:

- Configure the server to log messages to standard output, and route from there.
- Configure the server to log to files, and use log collection and routing for the log files.

ForgeRock provides support for using the following third-party software when monitoring ForgeRock servers:

Software	Version
Grafana	5 (at least 5.0.2)

Software	Version
Graphite	1
Prometheus	2.0

For hardware security module (HSM) support, ForgeRock software requires a client library that conforms to the PKCS#11 standard v2.20 or later.

FQDNs

Directory Services replication requires the use of fully qualified domain names (FQDNs).

Hostnames like `localhost` or `my-laptop.local` are acceptable for evaluation.

When setting up and configuring production servers, use FQDNs, and ensure DNS is set up correctly to provide FQDNs.

As a workaround when demonstrating across multiple host systems, you can update the hosts file (`/etc/hosts` or `C:\Windows\System32\drivers\etc\hosts`) to specify FQDNs.

Examples in the documentation use the hostname `localhost` to contact local DS servers. Trust in the examples depends on the use of a deployment key and password when setting up servers. A server certificate generated from a deployment key and password has `localhost` as the default hostname. By using the `--hostname localhost` option with a DS command-line tool, you simplify the secure connection process. When the tool validates the specified hostname against the hostname in the server certificate, they match. There is no need to add the server's hostname to the server certificate.

When making a secure connection to a *remote* server, be sure the FQDN in the `--hostname` *`fqdn`* option matches a valid hostname in the server certificate. If the server certificate is generated with a deployment key and password, you can easily renew the certificate to change or add a hostname. For examples, see "Replace a TLS Key Pair" in the *Security Guide* or "Generate a Key Pair (Wildcard Certificate)" in the *Security Guide*.

Adapt the examples as necessary when using your own certificates, keys, and PKI.

Clock Synchronization

Before using DS replication, set up synchronization between server system clocks.

To keep the system clocks synchronized, use a process that adjusts time to eventual clock consistency, such as `ntpd`. NTP adjusts the size of a second to move time to eventual clock consistency.

Once you have enabled replication, avoid moving the system clock in large increments, such as more than half a day at a time, or possibly less for systems under high load.

Certificates

For secure network communications with client applications that you do not control, 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 CA.

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 the server keystore, use the Java **keytool** command.

For details, see "Key Management" in the *Security Guide*.

Chapter 3
Incompatible Changes

+ *Accounts*

- The default directory superuser (Directory Manager) DN is now `uid=admin` for new servers.

  The upgrade process does not change the directory superuser DN for existing servers.

  This change makes it easier to manage the server configuration over REST, as the default identity mapper configuration maps the HTTP `admin` username to the LDAP DN `uid=admin`.

- The replication service discovery mechanism now obtains some information by reading the `cn=monitor` LDAP entry. As a result, the `bind-dn` account must now have the `monitor-read` privilege.

  This affects accounts used by DS directory proxy servers to bind to DS replication servers. For an example showing the account with the `monitor-read` privilege, see "Try DS Directory Proxy" in the *Installation Guide*.

+ *Backup*

- DS backups taken with this release are not compatible with backups from earlier releases.
- Scheduled backup tasks continue after upgrade.
- Tasks created with the **restore** command in earlier releases are removed during upgrade.

+ *Data*

The default backend ID for application data depends on the setup profiles.

The upgrade process does not change the backend ID for existing servers.

+ *Java APIs*

The server-side (plugin) Java API is continuing to evolve. See "*Interface Stability*".

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

+ *LDAP*

When matching strings in attributes with telephone number syntax, DS servers now behave as follows:

- As in previous versions, a search for `"(telephoneNumber=1555123456)"` matches entries with telephone number values `+1 555 123 456` and `1 555 123456`.
- All `+`s are ignored. In other words, `+` is no longer significant when matching a telephone number syntax attribute.
- A search for `"(telephoneNumber=*Flower*)"` returns only entries with telephone numbers containing `Flower` (case-insensitive match).
- A search for `"(telephoneNumber=15550102)"` no longer matches entries with telephone numbers like `+15550102 - Home`.

+ *Logging*

- The `batch` configuration for the JMS common audit handler for access logs has changed to support reconnection if the broker becomes unavailable.

  This change adds a `batch.writeInterval` setting. It removes the following settings:

  - `batch.batchEnabled`
  - `batch.insertTimeoutSec`
  - `batch.pollTimeoutSec`
  - `batch.shutdownTimeoutSec`
  - `batch.threadCount`

  For details on the JMS handler configuration, see "JMS" in the *Logging Guide*.
- The example JDBC audit handler configuration for logging to MySQL has changed.

  The old configuration is not compatible with MySQL 8, supported in this release.

  For details on the JDBC handler configuration, see "JDBC" in the *Logging Guide*.

+ *Mail*

The global property `smtp-server` has been replaced with a configuration object, "Mail Server".

+ *Replication*

- The server-id and group-id identifiers are now global settings, and only take a single value per server.

  Replication domain and replication server configurations no longer let you set `server-id` and `group-id` properties.

- The external changelog domain configuration has moved to the replication domain and replication server configurations. This affects the following properties:
  - ecl-include
  - ecl-include-for-deletes
  - changelog-enabled-excluded-domains
- The following replication domain configuration properties have moved to the replication synchronization provider:
  - changetime-heartbeat-interval
  - isolation-policy
  - heartbeat-interval
  - initialization-window-size
  - log-changenumber
  - referrals-url
  - solve-conflicts
  - source-address
- The following replication server properties have moved to the replication synchronization provider:
  - replication-purge-delay
  - source-address
- In addition to the property changes, the replication synchronization provider has changed:
  - A new property, bootstrap-replication-server, takes the addresses of one or more replication servers this server should contact to discover discover the rest of the topology.

- The replication-purge-delay property has replaced the replication domain property, `conflicts-historical-purge-delay`.

  In this release, the `replication-purge-delay` setting alone governs how long the replica retains data in the changelog and historical metadata necessary to solve conflicts in directory entries.

+ *REST*

The `resourceTypeProperty` field is no longer used in REST to LDAP configurations. The resource type is now inferred from the property with `"type": "resourceType"`.

+ *Security*

- Default security settings have been hardened.

  For details, see "Default Security Settings".

- The following configuration changes impact TLS-related settings:

  The "Crypto Manager" no longer has the following properties:

  - `ssl-cert-nickname`
  - `ssl-cipher-suite`
  - `ssl-encryption`
  - `ssl-protocol`

  The "Replication Synchronization Provider" configuration object now has the following properties:

  - `key-manager-provider`
  - `ssl-cert-nickname`
  - `ssl-cipher-suite`
  - `ssl-encryption`
  - `ssl-protocol`
  - `trust-manager-provider`

  The following configuration objects now have `ssl-cipher-suite` and `ssl-protocol` properties:

  - "HTTP OAuth2 OpenAM Authorization Mechanism"

- "HTTP OAuth2 Token Introspection (RFC 7662) Authorization Mechanism"
- "Replication Service Discovery Mechanism"
- "Static Service Discovery Mechanism"

- The default fingerprint algorithm for the fingerprint certificate mapper is now SHA-256.

+ *Setup*

The **setup** command has changed:

- The `--productionMode` option has been removed.

  Default settings are now secure. For details, see "Default Security Settings".

  The evaluation setup profile is compatible with other setup profiles. However, if you apply the evaluation setup profile last, it sets `unauthenticated-requests-policy:allow`, granting global permission to perform operations over insecure connections.

- Subcommands have been replaced by setup profiles.

- The **setup** command no longer starts the server by default.

  Before starting your new DS server, finish configuration. For details, see the examples in the Installation Guide.

  If no further configuration is required, use the **setup --start** option.

- For new servers, key pairs with self-signed certificates are no longer used. Instead, the setup process generates keys used for secure connections, and derives a shared master key to protect secret keys for data encryption. These keys depend on a deployment key and deployment key password. For details, see "Key Management" in the *Security Guide*.

  The deployment key and deployment key password are required as part of the setup process:

  - If you do not provide your own keys, and do not provide a deployment key, the **setup** command generates one for you. After it generates the key, the **setup** displays it in the command output.

  - If you do not provide your own keys, the generated keys and the signing CA certificate are stored in a PKCS#12 keystore file, `config/keystore`. The password is stored in a PIN file, `config/keystore.pin`. You can use the CA certificate as the root of trust for an entire deployment.

  - By default, replication now relies on the same key pairs as all other connection handlers to secure network communications.

The `Replication Key Manager` and `Replication Trust Manager` providers now point to the providers chosen during the setup process.

- The `Default Key Manager` is now named after its keystore format, such as `PKCS12`.

For details, see "Key Management" in the *Security Guide*.

- The following **setup** command options have been removed:
  - `-a, --addBaseEntry`
  - `-b, --baseDn`
  - `--useJvmTrustStore`
  - `-l, --ldifFile`
  - `-O, --doNotStart`
  - `--productionMode`
  - `-R, --rejectFile`
  - `--skipFile`

  Add your initial data before starting the server by creating a backend database, configuring indexes, and importing from LDIF. For details, see "*Data Storage*" in the *Configuration Guide*.
- The `-d, --sampleData` option has moved. It is now provided as the `generatedUsers` parameter of the `ds-evaluation` setup profile.

For examples using the command, see Installation Guide.

+ *Tools*

DS command line tools no longer support the `-w -` and `--bindPassword -` options to prompt interactively for a password.

Instead, provide the bind DN and omit the `-w -` or `--bindPassword -` option. The tools then prompt for a password unless you specify the `--no-prompt` option.

+ *Upgrade*

You can upgrade DS 3.0 and later servers directly to this release.

When starting from 2.6, *first upgrade all servers to DS 6.5* before upgrading further. Direct upgrade from 2.6 is no longer supported.

For details, see "Supported Upgrades" in the *Upgrade Guide*.

Default Security Settings

When you set up new DS servers, they are now configured with tighter security settings by default. These changes do not affect DS servers that you upgrade from earlier versions. If you require more lenient settings for compatibility, you must configure them after setting up the server:

- All operations except bind requests and StartTLS requests, and base object searches on the root DSE, require secure connections.

  This behavior is governed by the global configuration property, unauthenticated-requests-policy, which is now set to `allow-discovery`, instead of `allow`, unless the last setup profile applied is the `ds-evaluation` profile.

- The password storage scheme for the Default Password Policy and Root Password Policy is now `PBKDF2-HMAC-SHA256` with 10 iterations. For stronger security, raise the number of iterations as shown in "Configure a NIST-Inspired Subentry Policy" in the *Security Guide*, and require users to change their passwords.

  Warning

  `PBKDF2-HMAC-SHA256` is a computationally intensive one-way hashing scheme. When used with a high number of iterations, *it is intentionally orders of magnitude slower* than the previous default for user passwords, which was `Salted SHA-512`.

  `PBKDF2-HMAC-SHA256` and similar computationally intensive password storage schemes lower throughput and raise response times for some operations, including the following:

  - Importing plaintext passwords from LDIF; for example, during evaluation and testing with generated data.
  - Updating passwords.
  - Authenticating with passwords.

  Plan your deployment accordingly. For additional details, see "Password Storage" in the *Security Guide*.

  To migrate user passwords to a new storage scheme, see "Deprecate a Password Storage Scheme" in the *Security Guide*.

- SASL mechanism handler configurations for `CRAM-MD5` and `DIGEST-MD5` are no longer present in the default configuration.

- Password storage scheme configurations for `MD5`, `RC4`, and `Salted MD5` are no longer present in the default configuration.

  Less secure and reversible password storage schemes have been disabled in the default configuration. You must therefore enable these password storage schemes if you intend to use them.

Setting	New Default
Crypto Manager digest-algorithm	`SHA-256`
Crypto Manager key-wrapping-transformation	`RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING`
Crypto Manager mac-algorithm	`HmacSHA256`
Global setting unauthenticated-requests-policy	`allow-discovery`
Password storage scheme: 3DES enabled	`false`
Password storage scheme: AES enabled	`false`
Password storage scheme: Base64 enabled	`false`
Password storage scheme: Blowfish enabled	`false`
Password storage scheme: Clear enabled	`false`
Password storage scheme: CRYPT enabled	`false`
Password storage scheme: PBKDF2 enabled	`false`
Password storage scheme: PKCS5S2 enabled	`false`
Password storage scheme: Salted SHA-1 enabled	`false`
Password storage scheme: Salted SHA-256 enabled	`false`
Password storage scheme: Salted SHA-384 enabled	`false`
Password storage scheme: Salted SHA-512 enabled	`false`
Password storage scheme: SHA-1 enabled	`false`
Pluggable (JE) backend cipher-transformation	`AES/GCM/NoPadding`
Replication server cipher-transformation	`AES/GCM/NoPadding`

Chapter 4
Deprecation

The following are deprecated and likely to be removed in a future release:

- The previous format for password file options is deprecated. The options remain supported until removal, but are now hidden in online help. This affects the following options:

Deprecated Form	Use This Form
`--bindPasswordFile`	`--bindPassword:file`
`--deploymentKeyPasswordFile`	`--deploymentKeyPassword:file`
`--keyStorePasswordFile`	`--keyStorePassword:file`
`--monitorUserPasswordFile`	`--monitorUserPassword:file`
`--rootUserPasswordFile`	`--rootUserPassword:file`
`--trustStorePasswordFile`	`--trustStorePassword:file`

The following are deprecated since DS 7.0.0 and likely to be removed in a future release:

- Support for SNMP.

  DS software provides better options for monitoring servers, including support for Prometheus, Graphite, LDAP, and JMX. For details, see the Monitoring Guide.

  DS server software also includes a sample monitoring dashboard for Prometheus and Grafana, which is described in `opendj/samples/grafana/README.md`.

- The "pwdValidatorPolicy" object class.

  For subentry password policies, use the object classes derived from "ds-pwp-validator" instead.

- Reversible password storage schemes, and the `cn=admin data` base DN and `adminData` backend used to support them. This includes the following password storage schemes:

  - `3DES`
  - `AES`
  - `Blowfish`
  - `RC4`

- The HTTP monitoring endpoint, `/admin/monitor`.

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

- LDAP metrics:
  - `ds-mon-approx-oldest-change-not-synchronized`
  - `ds-mon-approximate-delay`
  - `ds-mon-missing-changes`
- Prometheus metrics:
  - `ds_replication_changelog_connected_replicas_approx_oldest_change_not_synchronized_seconds`
  - `ds_replication_changelog_connected_replicas_approximate_delay_seconds`
  - `ds_replication_changelog_connected_replicas_missing_changes`

Note

In mixed topologies, a directory server version 6 or earlier connected to a replication server version 6.5 or later cannot consume messages about other servers going offline. The monitoring framework reflects this as a delay on the directory server that could not consume the message.

The delay is calculated correctly again once all servers in the topology are upgraded to at least version 6.5, or when the offline server comes back online and has seen a change to directory data.

Monitor replication delay instead of using the deprecated metrics. For details, see "Replication Delay (LDAP)" in the *Monitoring Guide* or "Replication Delay (Prometheus)" in the *Monitoring Guide*.

Chapter 5
Removed

- Support for Java 8 has been removed.

  Support for 32-bit JVMs has also been removed.

  When upgrading to this version, follow the instructions in "Supported Java" in the *Upgrade Guide*.

- The **backup** and **restore** commands have been removed. Use the **dsbackup** command instead.

- The **dsreplication** command has been removed.

  You now configure replication as part of the setup process using the **setup --replicationPort** and **setup --bootstrapReplicationServer** options. For details and examples, see the Installation Guide.

  For most operations, use the **dsrepl** command. Since replication configuration is part of the setup process, the **dsrepl** command does not include a command for configuring replication. For examples using the new command, see "*Replication*" in the *Configuration Guide*, and "*Changelog for Notifications*" in the *Configuration Guide*.

  To temporarily suspend and resume replication, use the **dsconfig** command. For details, see "Disable Replication" in the *Configuration Guide*.

- The `ads-truststore` and `ads-truststore.pin` files have been removed.

  For new deployments, DS servers protect secret keys with a shared master key. The setup process derives the shared master key from the deployment key and password.

- The JVM profiler plugin has been removed in this release.

- The following monitoring metrics depending on the JVM implementation are not stable interfaces. They have been removed from the documentation:

  Garbage collection statistics

  Affected metrics have names like `ds-mon-jvm-garbage-collector-*` under `cn=monitor`, and `ds_jvm_garbage_collector_*` in Prometheus output.

  Memory pool use

  Affected metrics have names like `ds-mon-jvm-memory-pools-*` under `cn=monitor`, and `ds_jvm_memory_pools_*` in Prometheus output.

- The `No-Op` alias for the LDAP no-op control (OID: `1.3.6.1.4.1.4203.1.10.2`) has been removed.

Use the `NoOp` alias or the OID instead.

Chapter 6

Fixes

Fixed in 7.0.2

The following important bug was fixed in DS 7.0.2:

- OPENDJ-7810: JMX connections are always considered insecure

Fixed in 7.0.1

The following important bugs were fixed in DS 7.0.1:

- OPENDJ-7674: Migrating encrypted changelog files during upgrade fails
- OPENDJ-7612: replication divergence on CTS in the cloud
- OPENDJ-7599: Cannot add a pre-encoded password to an entry without an existing password
- OPENDJ-7554: Windows: Secrets not retrieved from :file command-line arguments
- OPENDJ-7450: The startswith (sw) operator on indexed JSON attribute is slow
- OPENDJ-7443: AM Identity Store 7.0 Setup profile missing "push2faEnabled" attribute
- OPENDJ-7436: Backup to the cloud takes too much time
- OPENDJ-5927: Server stuck on a DS trying to reconnect to an RS
- OPENDJ-7523: Example plugin and example pwdscheme pom.xml is missing 7.0.0 as revision

Fixed in 7.0.0

The following important bugs were fixed in DS 7.0.0:

- OPENDJ-6499: Query on rest2ldap over ssl gets stuck after few curl requests using TLSv1.3 on JDK11
- OPENDJ-7115: DS does not start when deployed with ISTIO side car container in the GCP K8s cloud
- OPENDJ-640: Text Query Against indexed telephoneNumber Attribute Very Slow

- OPENDJ-6235: Stale ds-sync-hist attribute values reappear in the entry after replication is unconfigured
- OPENDJ-6512: Problems when work queue fills
- OPENDJ-6778: Proxy server mishandles abandon requests
- OPENDJ-6787: Changelog searches are extremely slow if any cursors are exhausted
- OPENDJ-6221: Logging for CONNECT operations are not saved in Nanosecond format
- OPENDJ-6196: HTTP connection handler continues to listen to 0.0.0.0 after setting listen-address
- OPENDJ-6222: SMTP messages are sometimes not encoded with the correct charset
- OPENDJ-6240: DS not honoring per user resource limits when processing RESTful operation requests
- OPENDJ-6116: Unspecified Communications Error when multiple rest2ldap endpoints share configuration elements
- OPENDJ-6527: Server does not return password policy responses with only warnings
- OPENDJ-6557: IDM Password Sync plugin induces 100% CPU in Apache Http Components when used with JDK 11
- OPENDJ-5661: supportextract tool help and version options are different from other tools
- OPENDJ-5675: JDK11: supportextract tool cannot find jstack command
- OPENDJ-5664: JDK 11: illegal reflective access warning during import-ldif
- OPENDJ-5590: Proxy: server discovery fails silently when proxy base-dn differs from backend's base-dn
- OPENDJ-5584: Server does not validate sum of memory used by JE backend caches in all cases
- OPENDJ-4764: REST2LDAP gateway sasl-plain authorization doesn't handle dn: correctly
- OPENDJ-6188: Backend returns an incorrect error type when disk space hits low threshold
- OPENDJ-6349: "RuntimeException: Should never happen" in HttpClientConnection
- OPENDJ-5660: JDK 11: illegal reflective access warning on setup (with profile)
- OPENDJ-6540: The Supportextract hangs when loggers are configured to use /dev/stdout
- OPENDJ-6711: Replication status reports The provided value "5277383431" could not be parsed as an integer.
- OPENDJ-7016: status command outputs malformed JSON in script friendly mode

- OPENDJ-5611: Change number indexing can lag behind replication under extreme load
- OPENDJ-6173: cn=monitor memory pool stats do not get updated properly over time
- OPENDJ-6377: Replication replay: issues with ReplaySynchronizer
- OPENDJ-7176: Filters with malformed attribute descriptions cannot be parsed
- OPENDJ-6733: SMTP handler sends incorrect email when account status is modified by manually updating ds-pwp-account-disabled attribute
- OPENDJ-6521: setup checks admin port despite options --skipPortCheck --doNotStart
- OPENDJ-4714: SSL handshake now sends 16KB list of CA issuer DNs
- OPENDJ-7319: Addrate can run out of memory when --deleteMode off and --noPurge are set
- OPENDJ-7286: Changelog searches can start with incorrect cursors
- OPENDJ-6994: strict-format-country-string does not affect the server
- OPENDJ-3121: Setup fails to create the lib/extensions directory in the instance.loc path, if a instance.loc file is used.
- OPENDJ-2605: Debian packages should be idempotent
- OPENDJ-1169: Exception/error lost when logging ERR_LOOP_REPLAYING_OPERATION

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 7
Limitations

These limitations are inherent to the design, not bugs to be fixed:

Category	Limitations
Account Lockout	When you configure account lockout as part of password policy, DS servers lock an account after the specified number of consecutive authentication failures. *Account lockout is not transactional across all replicas in a deployment.* Global account lockout occurs as soon as the authentication failure times have been replicated.
LDAP	• DS servers provide full LDAP v3 support, except for alias dereferencing, and limited support for LDAPv2. • 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.
Passwords	• Directory servers store passwords prefixed with the storage scheme in braces, as in `{scheme}`. 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 that passwords may not be provided in pre-encoded form. • 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.
Proxy Server	• Configuring a server with both local backends and proxy backends is not supported. Access control models for directory servers and proxy servers do not function at the same time in the same server. • 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.

Category	Limitations

	• When using ACIs or collective attributes with the proxy server data distribution feature, the ACI and entries having collective attribute values must be located at or above the `partition-base-dn`. When changing this data, make the change behind the proxy to one directory server replica in each shard. Your changes are not replicated outside the shard.
	
	The proxy server data distribution feature does not currently support the following:
	
	• Importing distributed data with the **import-ldif** command.
	
	• Changes to the number of partitions after data has been deployed.
	
	• Modify DN operations to distributed entries.
	
	• Updates to entries at or above the `partition-base-dn`.
	
	• Virtual static groups.
	
	• Data distribution does not support these virtual attributes:
	
	`member`
	`uniqueMember`
	
	The `isMemberOf` virtual attribute works as expected as long as you replicate the group entries on every shard.
	
	• Data distribution does not support these LDAP controls:
	
	Server-Side Sort controls: `1.2.840.113556.1.4.473`, `1.2.840.113556.1.4.474`
	Simple Paged Results control: `1.2.840.113556.1.4.319`
	Virtual List View controls: `2.16.840.1.113730.3.4.9`, `2.16.840.1.113730.3.4.10`

REST to LDAP	• 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. 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 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, perform an update of the entire object, changing only the desired fields in your copy.
Windows	Due to a Java issue on Windows systems (JDK-8057894), when configuring DS servers with data confidentiality enabled, you might see an error message containing the following text: `Unexpected CryptoAPI failure generating seed`

Category	Limitations
	If this happens, try running the command again.

Chapter 8

Known Issues

The following important issues remained open at the release of 7.0.2:

- OPENDJ-7905: Schema replication error after upgrade
- OPENDJ-7474: Docker sample README.md provides wrong instructions for running the container
- OPENDJ-7689: dsrepl add-local-server-to-pre-7-0-topology does not tolerate separate keystore and truststore
- OPENDJ-7643: Log that is supposedly generated from dsreplication operation is empty or does not exist
- OPENDJ-7516: External cn=changelog is not updated while dsreplication initialization is in progress
- OPENDJ-7654: DS unable to connect to RS after full gc
- OPENDJ-7763: Proxy service discovery with RS-only and DS-only seems not to route search
- OPENDJ-7744: dsrepl initialize in a topology with DS7 and DS 5.5 fails if DS7 serverId starts with 0
- OPENDJ-7837: Schema replication issues can result in duplicate schema and out-of-sync schema
- OPENDJ-7818: Package based upgrade should be done as the root user
- OPENDJ-7788: dsrepl initialize causes the ReplicationDomain listener to die with an NPE
- OPENDJ-7755: DS 7.0 replication with older version, CryptoManager failed to import the symmetric key entry
- OPENDJ-7851: Supportextract tool: clobbers the server.out filehandle when kill -3 is used.
- OPENDJ-7513: Missing subSchemaSubEntry attribute from rootDSE access controls
- OPENDJ-7844: Many limitations encountered trying to override core schema
- OPENDJ-7481: JSON logs do not contain proxy auth DN
- OPENDJ-7747: ldapmodify display full stack exception on LDIF errors if connection is already established
- OPENDJ-7816: dsbackup fails when destination is a symbolic link to a real directory

- OPENDJ-7737: ConfigurationFramework#initialize0 changes the class loader without clearing the map of registered jar files
- OPENDJ-7847: StaticGroup's objectclass sanity checks are unhelpful
- OPENDJ-7758: DS 7.0 dsrepl add-local-server-to-pre-7-0-topology: NPE if master-key is in different keystore
- OPENDJ-7655: Replaying multiple MODIFYDN operations is very slow
- OPENDJ-7867: NPE while backing up to GCS
- OPENDJ-7699: Supportextract throws NoSuchElementException when the server.pid file is empty
- OPENDJ-7687: Global Access Control Policy regarding cn=schema is too restrictive
- OPENDJ-7745: DS6.5 - 7 Replication fails in Kubernetes
- OPENDJ-7841: S3 Backup fails with low resources

Chapter 9
Documentation

Date	Description
2021-04-06	• Updated "Restore" in the *Maintenance Guide* to clarify that even a single directory server replica replays changes after you restore data, and to link to the procedure for restoring data to a known state, should you choose to prevent this. • Added "Remove a Bootstrap Replication Server" in the *Configuration Guide* to demonstrate how to purge a bootstrap replication server from other servers' configurations. DS servers now purge replica state from memory and from changelogs when a replica disappears for longer than the replication purge delay. This happens without any administrative action. They do not purge bootstrap replication servers from their configurations, however. You must do that manually after retiring a bootstrap replication server. • Updated "*Known Issues*" to mention the issue OPENDJ-7905. Follow the link to the issue to find suggested workarounds.
2021-03-29	Release of Directory Services 7.0.2 software.
2021-02-04	Adapted the Upgrade Guide to make it easier to use: • "*When Adding New Servers*" in the *Upgrade Guide* reiterates that replication configuration now happens at setup time, and clarifies that the new replica cannot initially connect to an existing standalone replication server. The existing server must be a directory server, not a standalone replication server. • "*After You Upgrade*" in the *Upgrade Guide* describes how to manually update LDAP schema after upgrade.
2020-12-10	Release of Directory Services 7.0.1 software. The following changes were made to the documentation: • Updated "*When Adding New Servers*" in the *Upgrade Guide* to reiterate that replication configuration now happens at setup time. • Updated "*Install DS for User Data*" in the *Installation Guide* to clarify that the procedure is for installing your own user data. • Updated "Restrict Protocols and Cipher Suites" in the *Security Guide* to correct the examples demonstrating how to configure a server to require TLSv1.3.

Date	Description
	• Updated "List Protocols and Cipher Suites" in the *Security Guide* to correct the description of the `ECDHE_RSA` algorithm. • Updated "Use a Non-Default Superuser Account" in the *Security Guide* to add the missing `cn` attribute in the sample alternative directory superuser entry. • Updated "Use the Debian Package" in the *Installation Guide* to demonstrate installing the `java11-runtime` virtual package, which is a dependency of the DS Debian package.
2020-09-07	Fix Javadoc search box.
2020-08-10	Initial release of Directory Services 7 software. In addition to the changes described elsewhere in these notes, the following important changes were made to the documentation: **Best Practices** • Updated "Java Settings" in the *Maintenance Guide* to reflect current recommendations for JVM settings. • Added "Linux Page Caching" in the *Maintenance Guide* to explain how to avoid long pauses when the kernel flushes dirty pages to disk. • Added "Disaster Recovery" in the *Maintenance Guide*. • Added "*On Load Balancers*" in the *Configuration Guide* with recommendations for your deployment. • Added "Tune Settings" in the *Upgrade Guide* to underline the importance of revisiting tuning settings during major version upgrades. • Added "Custom Schema" in the *Configuration Guide* to demonstrate how to add a custom attribute with an enumeration syntax. • Updated "Server Commands" in the *Maintenance Guide* to explain that commands that change server files must run as a user who has the same filesystem permissions as the user who installs and runs the server. • Updated "*Groups*" in the *Configuration Guide* to more strongly recommend dynamic groups over static groups. • Updated "*Authentication (Binds)*" in the *LDAP User Guide* to emphasize the role of identity mappers. **Better Examples** • Many more examples throughout the documentation now use secure connections to servers. Examples in the documentation use the hostname `localhost` to contact local DS servers. Trust in the examples depends on the use of a deployment key and password when setting up servers. A server certificate generated from a deployment key and password has `localhost` as the default hostname. By using

Date	Description
	the `--hostname localhost` option with a DS command-line tool, you simplify the secure connection process. When the tool validates the specified hostname against the hostname in the server certificate, they match. There is no need to add the server's hostname to the server certificate. When making a secure connection to a *remote* server, be sure the FQDN in the `--hostname` *`fqdn`* option matches a valid hostname in the server certificate. If the server certificate is generated with a deployment key and password, you can easily renew the certificate to change or add a hostname. For examples, see "Replace a TLS Key Pair" in the *Security Guide* or "Generate a Key Pair (Wildcard Certificate)" in the *Security Guide*. Adapt the examples as necessary when using your own certificates, keys, and PKI. • Updated Getting Started to include Windows PowerShell examples alongside the Bash examples. • Updated "*HTTP-Based Monitoring*" in the *Monitoring Guide* and "*LDAP-Based Monitoring*" in the *Monitoring Guide* to add examples showing how to monitor operation and work queue statistics, database size, and active users. **Errata** • Updated "*Install DS for AM Identities*" in the *Installation Guide* to mention that the default base DN for the profile is `ou=identities`. • Updated "*Metric Types Reference*" in the *Monitoring Guide* to clarify the definitions of monitoring metrics. • Updated "Active Accounts" in the *LDAP User Guide* to fix an incorrect example, and to clarify that the format string must match the syntax of the attribute. • Updated "Necessary Indexes" in the *Configuration Guide* to use the appropriate filter use metrics, which were new in DS 6. • Updated "*Support for Languages and Locales*" in the *LDAP Reference* to clarify that although DS software supports many locales, DS software is only partially localized. **Reorganization** • The documentation now uses titles that better reflect the content. For details, see *Having Trouble Finding Something?*. • Reworked "*Install Directory Proxy*" in the *Installation Guide* to make it easier to get started with DS directory proxy servers. • Moved the section on configuration expressions to "*Property Value Substitution*" in the *Configuration Reference*. • Updated "Enable the External Changelog" in the *Configuration Guide* to clarify which configuration is required, and to cover standalone directory servers.

Date	Description
	The separate procedure, *To Enable the External Change Log (Standalone Server)*, has been removed. **Replication** • Added "Port Use and Operations" in the *Configuration Guide* to describe how replication uses DS server ports that must remain open to remote clients. • Updated "Disk Space Thresholds" in the *Configuration Guide* to remove the claim that replication updates are applied after the disk full threshold is reached. • Updated "*When Adding New Servers*" in the *Upgrade Guide* to explain that you must set up replication for the first time between servers of the same version. • Updated "*Install Standalone Servers*" in the *Installation Guide* and "*Replication*" in the *Configuration Guide* to clarify that if you want to restrict TLS protocols or cipher suites, do so before configuring replication. • Updated "About Replication" in the *Configuration Guide* and "Clock Synchronization" to clarify how replication is resilient to host clock anomalies. • Updated "Replication Delay (LDAP)" in the *Monitoring Guide* and "Replication Delay (Prometheus)" in the *Monitoring Guide* to provide a better explanation of replication delay metrics. • "*Install Standalone Servers*" in the *Installation Guide* demonstrates simplified setup processes. Redundant procedures for standalone servers were removed from the documentation on configuring replication. • Simplified "*Move a Server*" in the *Maintenance Guide*. **REST** • Added "Map LDAP Entries" in the *HTTP User Guide* and "Nested Resources" in the *HTTP User Guide* to demonstrate additional REST to LDAP mappings. • Updated "API Configuration" in the *HTTP User Guide* to clarify that resource type names can only be reused when referring to identical resource type definitions. **Security** • Added "Authenticate With a Third-Party Certificate" in the *Security Guide* to clarify how to safely allow authentication with client certificates that are self-signed or not signed by a CA you control. • Updated "Certificate-Based Authentication" in the *Security Guide* to clearly indicate how to trust a certificate that was not signed by a well-known CA. • Updated "Assign Password Policies" in the *Security Guide* to clarify that you must not assign more than one password policy to the same account. • Rewrote parts of "*Cryptographic Keys*" in the *Security Guide* to clarify when to use a private CA and when to use a public CA.

Date	Description
	Setup • Added a reference to the setup profile API in "Create Your Own" in the *Installation Guide*. • Added a table of backends with default settings to "*Data Storage*" in the *Configuration Guide*.

Chapter 10

Interface Stability

Interfaces labelled as Evolving in the documentation may change without warning. 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.
- Text in log messages should be considered Internal. Log message IDs are Evolving.
- The default content of `cn=schema` (LDAP schema) is Evolving.
- The interface of the **changelogstat** command is Evolving.
- 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`, is Internal.
- Also see "*Deprecation*" and "*Removed*".

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

Release Label	Version Numbers	Characteristics
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

ForgeRock Product Stability Labels

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

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

ForgeRock Stability Label Definitions

Stability Label	Definition
Stable	This documented feature or 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 feature or 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.
Legacy	This feature or interface has been replaced with an improved version, and is no longer receiving development effort from ForgeRock. You should migrate to the newer version, however the existing functionality will remain.

Stability Label	Definition
	Legacy features or interfaces will be marked as *Deprecated* if they are scheduled to be removed from the product.
Deprecated	This feature or interface is deprecated and likely to be removed in a future release. For previously stable features or interfaces, the change was likely announced in a previous release. Deprecated features or interfaces will be removed from ForgeRock products.
Removed	This feature or 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 considered as new technology that is 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 features or interfaces can change without notice. If you depend on one of these features or interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Appendix A. Getting Support

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.

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.