Installation Guide

/ OpenDJ 3.5

Latest update: 3.5.3

Mark Craig

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

Abstract

This guide shows you how to install OpenDJ directory services. The OpenDJ project offers open source LDAP directory services in Java.

Table of Contents

Preface

This guide shows you how to install, upgrade, and remove OpenDJ software. Unless you are planning a throwaway evaluation or test installation, read the Release Notes before you get started.

If you only want to try OpenDJ server software, and you do not plan to store any real or important data that you want to keep, then you need not read this entire guide. Instead read "To Prepare For Installation" and "To Install OpenDJ Directory Server With the GUI".

1. Who Should Read this Guide

This guide is written for anyone installing OpenDJ who plans to maintain directory services for client applications. Basic OpenDJ installation can be simple and straightforward, particularly if you are already acquainted with directory services. Upgrading a running directory service without a single point of failure that can cause downtime requires at least a little thought and planning. If you are doing a basic installation, you might find yourself wanting more information about the process.

This guide covers the install, upgrade, and removal (uninstall) procedures that you theoretically perform only once per version. This guide aims to provide you with an understand of what happens when you perform the steps.

You do not need to be an LDAP wizard to learn something from this guide, though knowing how to manage directory services helps. You do need to know how to manage servers and services on your operating system of choice. You can nevertheless get started with this guide, and then learn more as you go along.

2. Formatting Conventions

Most examples in the documentation are created in GNU/Linux or Mac OS X operating environments. If distinctions are necessary between operating environments, examples are labeled with the operating environment name in parentheses. To avoid repetition file system directory names are often given only in UNIX format as in `/path/to/server`, even if the text applies to `C:\path\to\server` as well.

Absolute path names usually begin with the placeholder `/path/to/`. This path might translate to `/opt/`, `C:\Program Files\`, or somewhere else on your system.

Command-line, terminal sessions are formatted as follows:

```
$ echo $JAVA_HOME
/path/to/jdk
```

Command output is sometimes formatted for narrower, more readable output even though formatting parameters are not shown in the command.

Program listings are formatted as follows:

```
class Test {
    public static void main(String [] args)  {
        System.out.println("This is a program listing.");
    }
}
```

3. Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

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

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

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

4. Using the ForgeRock.org Site

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

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

Chapter 1
Installing OpenDJ Servers

This chapter covers installation of OpenDJ server software and includes the following procedures:

To Prepare For Installation

1. Make sure you have a required Java environment installed as described in "Java Environment" in the *Release Notes*.

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

2. Prevent antivirus and intrusion detection systems from interfering with OpenDJ directory server.

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

3.

Download enterprise software releases through the ForgeRock BackStage site. ForgeRock enterprise releases are thoroughly validated builds for ForgeRock customers who run OpenDJ in production deployments, and for those who want to try or test with release builds.

The following OpenDJ 3.5.3 server software is available:

opendj-3.5.3.zip
opendj-oem-3.5.3.zip (OEM Edition)

Cross-platform OpenDJ directory server installation files.

opendj_3.5.3-1_all.deb
opendj-oem_3.5.3-1_all.deb (OEM Edition)

OpenDJ directory server native package for Debian and related Linux distributions.

opendj-3.5.3-1.noarch.rpm
opendj-oem-3.5.3-1.noarch.rpm (OEM Edition)

OpenDJ directory server native package for Red Hat and related Linux distributions.

opendj-dsml-servlet-3.5.3.war

Cross-platform OpenDJ DSML gateway web archive

opendj-rest2ldap-servlet-3.5.3.war

Cross-platform OpenDJ REST to LDAP gateway web archive

Note

The OEM distribution of OpenDJ directory server does not include Berkeley DB Java Edition, and so does not support JE backends.

4. If you plan to install OpenDJ DSML gateway or OpenDJ REST to LDAP gateway, make sure you have an appropriate application server installed.

   For a list of supported application servers, see "Application Servers" in the *Release Notes*.

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

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

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

To Install OpenDJ Directory Server With the GUI

The OpenDJ **setup** command launches a wizard that lets you install OpenDJ directory server through a GUI.

Note

If your environment picks up an old installation of Java, installation can fail. You might see an application error due to an old Java version.

After completing the steps in "To Prepare For Installation", follow these steps:

1. Unzip opendj-3.5.3.zip, and then run the **setup** command, described in setup(1) in the *Reference*.

   When you unzip `opendj-3.5.3.zip`, a top-level `opendj` directory is created in the directory where you unzipped the file. On Windows systems if you unzip `opendj-3.5.3.zip`, with Right-Click > Extract All, be sure to remove the trailing `opendj-3.5.3` directory from the folder you specify.

   Find the **setup** command in the following locations:

   - (UNIX|Linux) **opendj/setup**
   - (Windows) **opendj\setup.bat**

2. Follow the instructions in the wizard.

   The wizard presents the following screens:

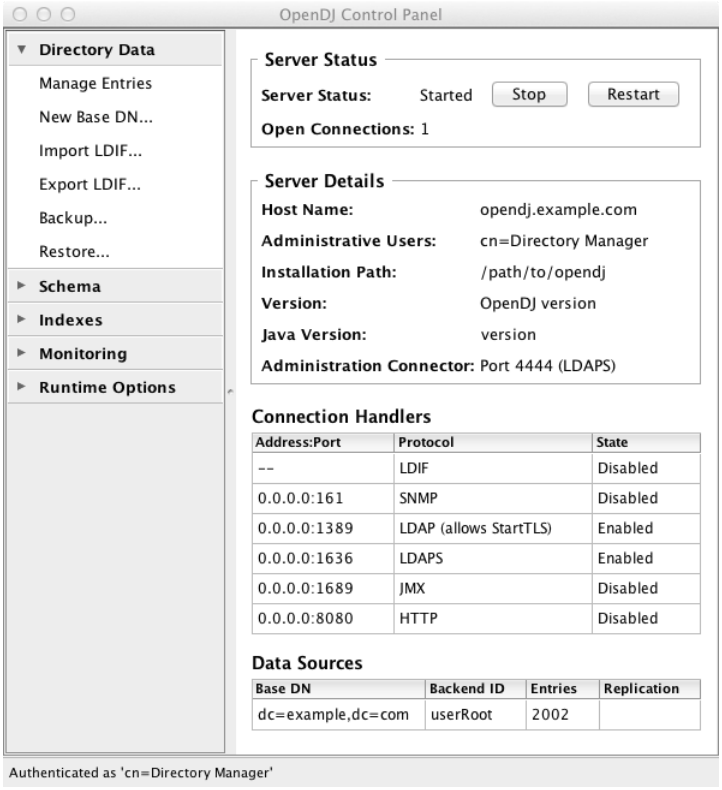
   - *Welcome*: summarizes the setup process and indicates the minimum required Java version.
   - *License*: presents the license agreement to accept before installing OpenDJ software.
   - *Server Settings*: prompts for basic server settings including installation path, host name, port numbers, secure connections, and credentials for the directory superuser (default bind DN: `cn=Directory Manager`).
   - *Topology Options*: prompts for data replication options including whether this server is part of a replication topology, and if so, the port number and security settings for this server, as well as the connection settings for a remote replica, if available.
   - *Directory Data*: allows you to import or to generate LDAP directory data as part of the setup process.

     This screen also allows you to select the backend type for data storage.

- *Runtime Options*: allows you to adjust JVM settings as part of the setup process, for example, to allow OpenDJ to use more memory if necessary.
- *Review*: presents current selections so that you can check everything is correct before running setup, with the option to start OpenDJ directory server after setup completes.
- *Finished*: summarizes how setup completed, with the option to launch the OpenDJ control panel.

"OpenDJ Control Panel" shows the top-level window with status information. OpenDJ control panel manages directory data, LDAP schema, indexes, monitoring, and JVM runtime options through a GUI.

OpenDJ Control Panel



To Start OpenDJ Control Panel

You might close OpenDJ control panel, or decide to start it later after closing the setup wizard:

- To launch OpenDJ control panel, run the **control-panel** command, described in control-panel(1) in the *Reference*.

  Depending on your host system, this command is one of the following:

  - (Linux|UNIX) **/path/to/opendj/bin/control-panel**
  - (Windows) **C:\path\to\opendj\bat\control-panel.bat**

To Separate OpenDJ Directory Server Tools From Data

The OpenDJ directory server **setup** command starts with OpenDJ tools and libraries distributed with the software, and generates the configuration files, log files, and data files required to run the server and to hold directory data. By default, all the files are co-located. Optionally, you can choose to put the data files in a different location from the tools and server libraries. After OpenDJ server tools and libraries are installed, but before the **setup** command is run, an `instance.loc` file can be used to set a different location for the configuration, logs, and data files.

Important

You cannot use a single set of server tools for multiple servers.

Tools for starting and stopping the server process, for example, work with a single configured server. They do not have a mechanism to specify an alternate server location.

If you want to set up another server after running the **setup** command, install another set of tools and libraries.

Follow these steps to put the configuration, logs, and data files in a different location:

1. Before running the **setup** command, create an `instance.loc` file to identify the location.

   The **setup** command tries to read `instance.loc` in the same directory as the **setup** command, such as `/path/to/opendj/`.

   The `instance.loc` file contains a single line identifying either the absolute location, such as `/path/to/server`, or the location relative to the `instance.loc` file.

2. Run the **setup** command to complete OpenDJ directory server installation.

   The directories for the server configuration, logs, and data files are located in the directory identified in the `instance.loc` file.

To Install OpenDJ Directory Server From the Command-Line

The OpenDJ **setup --cli** command launches a command-line installation that is interactive by default. After completing the steps in "To Prepare For Installation", follow these steps:

1. Unzip `opendj-3.5.3.zip` in the file system directory where you want to install the server.

The **setup** command, described in setup(1) in the *Reference*, uses the directory where you unzipped the files as the installation directory, and does not ask you where to install OpenDJ directory server. Therefore, if you want to install elsewhere on the file system, unzip the files in that location.

When you unzip `opendj-3.5.3.zip`, a top-level `opendj` directory is created in the directory where you unzipped the file. On Windows systems if you unzip `opendj-3.5.3.zip`, with Right-Click > Extract All, be sure to remove the trailing `opendj-3.5.3` directory from the folder you specify.

2. Run the **setup --cli** command found in the `/path/to/opendj` directory.

   This command starts the setup program in interactive mode on the command-line, prompting you for each option. Alternatively, use additional **setup** options to specify values for the options you choose during interactive mode, thus scripting the installation process. See **setup --help** and the notes below.

   To perform a non-interactive, silent installation, provide all the options to configure OpenDJ, and then also use the `-n` or `--no-prompt` option.

   The **setup** command without the `--cli` option runs the GUI installer.

   The following example shows interactive installation of OpenDJ directory server:

```
    $ /path/to/opendj/setup --cli
READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING
THE FORGEROCK SOFTWARE, YOU, ON BEHALF OF YOURSELF AND YOUR COMPANY, AGREE TO
BE BOUND BY THIS SOFTWARE LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THESE
TERMS, DO NOT DOWNLOAD OR INSTALL THE FORGEROCK SOFTWARE.

...

Please read the License Agreement above.
You must accept the terms of the agreement before continuing with the
installation.
Accept the license (Yes/No) [No]:Yes

What would you like to use as the initial root user DN for the Directory
Server? [cn=Directory Manager]:
Please provide the password to use for the initial root user:
Please re-enter the password for confirmation:

Provide the fully-qualified directory server host name that will be used when
generating self-signed certificates for LDAP SSL/StartTLS, the administration
connector, and replication [opendj.example.com]:

On which port would you like the Directory Server to accept connections from
LDAP clients? [1389]:

On which port would you like the Administration Connector to accept
connections? [4444]:

Do you want to create base DNs in the server? (yes / no) [yes]:
```

```
Provide the backend type:

    1)  JE Backend
    2)  PDB Backend

Enter choice [1]: 2

Provide the base DN for the directory data: [dc=example,dc=com]:

Options for populating the database:

    1)  Only create the base entry
    2)  Leave the database empty
    3)  Import data from an LDIF file
    4)  Load automatically-generated sample data

Enter choice [1]: 3

Please specify the path to the LDIF file containing the data to import:
/path/to/Example.ldif

Do you want to enable SSL? (yes / no) [no]:

Do you want to enable Start TLS? (yes / no) [no]:

Do you want to start the server when the configuration is completed? (yes /
no) [yes]:


Setup Summary
=============
LDAP Listener Port:            1389
Administration Connector Port: 4444
JMX Listener Port:
LDAP Secure Access:            disabled
Root User DN:                  cn=Directory Manager
Directory Data:                Create New Base DN dc=example,dc=com.
Base DN Data: Import Data from LDIF File (/path/to/Example.ldif)

Start Server when the configuration is completed


What would you like to do?

    1)  Set up the server with the parameters above
    2)  Provide the setup parameters again
    3)  Print equivalent non-interactive command-line
    4)  Cancel and exit

Enter choice [1]:

See /var/.../opendj-setup...log for a detailed log of this operation.

Configuring Directory Server ..... Done.
Importing LDIF file /path/to/Example.ldif ........... Done.
Starting Directory Server ........... Done.

To see basic server configuration status and configuration you can launch \
/path/to/opendj/bin/status
```

Notes on the options follow:

Initial root user DN

The root user Distinguished Name (DN) identifies a user who can perform all operations allowed for the server, called root user due to the similarity to the UNIX root user.

The default, `cn=Directory Manager`, is a well-known name. For additional protection, use a different name.

Initial root user password

The root user will use simple, password-based authentication. Later you can limit cleartext access to avoid snooping, but for now use a strong password here unless this is a throwaway server.

Fully qualified directory server host name

OpenDJ uses fully qualified host name in self-signed certificates and for identification when you use replication.

If you are installing a single server temporarily for evaluation, and are not concerned about replication and whether self-signed certificates can be trusted, then you can use an FQDN such as `localhost.localdomain`.

Otherwise, use an FQDN that other hosts can resolve to reach your server.

LDAP port

The default for LDAP is 389.

If you are working as a user who cannot open port 389, setup suggests 1389 by default.

Administration port

The default is 4444.

This is the service port used to configure the server and to run tasks.

Create base DNs

You need a base DN, such as `dc=example,dc=com`, to add directory data. If you already have LDIF, the base DN you want is the DN suffix common to all entries in your LDIF.

When you choose to create a base DN, the **setup** command also prompts you for a backend type, which identifies the implementation of the repository that holds your data.

Later you can add more base DNs if your data belongs in more than one suffix.

Import LDIF

LDAP data interchange format (LDIF) is the standard text format for expressing LDAP data.

If you have LDIF already, one reason you might not want to import the data right away is because your data uses attributes not defined in the default schema. Add schema definitions after installation, and then import from LDIF.

If you have a large data set to import, also increase the import cache size, which you can do by passing a Java properties file. You might also prefer to perform data import offline.

Enable SSL and TLS

Enabling SSL or TLS lets you protect the network traffic between directory clients and your server:

SSL

SSL requires its own, separate port for LDAPS traffic.

The default port for LDAPS is 636.

If you are working as a user who cannot open port 636, setup suggests 1636 by default.

TLS

TLS lets you use StartTLS to negotiate a secure connection between a client and server, starting from the same server port you configured for LDAP.

X.509 certificates

The digital certificate you need for SSL and TLS can be self-signed and created while you are working. Remember that client applications view self-signed certificates like fake IDs, and so do not trust them.

Self-signed certificates for externally facing ports facilitate testing, but are not intended for production use.

Start the server

If you do not start the server during installation, you can use the **/path/to/opendj/bin/start-ds** command later.

3. Run the **status** command, described in status(1) in the *Reference*, to make sure your OpenDJ server is working as expected as shown in the following example:

```
$ /path/to/opendj/bin/status

>>>> Specify OpenDJ LDAP connection parameters
```

```
Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

          --- Server Status ---
Server Run Status:        Started
Open Connections:         1

          --- Server Details ---
Host Name:                opendj.example.com
Administrative Users:     cn=Directory Manager
Installation Path:        /path/to/opendj
Version:                  OpenDJ 3.5.3
Java Version:             version
Administration Connector: Port 4444 (LDAPS)

          --- Connection Handlers ---
Address:Port : Protocol : State
-------------:----------:--------
-
--           : LDIF     : Disabled
0.0.0.0:161  : SNMP     : Disabled
0.0.0.0:636  : LDAPS    : Disabled
0.0.0.0:1389 : LDAP     : Enabled
0.0.0.0:1689 : JMX      : Disabled

          --- Data Sources ---
Base DN:     dc=example,dc=com
Backend ID:  userRoot
Entries:     160
Replication: Disabled
```

Note

You can install OpenDJ in unattended and silent fashion, too. See the procedure, "To Install OpenDJ Directory Server With a Properties File".

To Install From the Debian Package

On Debian and related Linux distributions such as Ubuntu, you can install OpenDJ directory server from the Debian package:

1. (Optional) Before you install OpenDJ, install a Java runtime environment if none is installed yet:

```
$ sudo apt-get install default-jre
```

2. Install the OpenDJ directory server package:

```
$ sudo dpkg -i opendj_3.5.3-1_all.deb
Selecting previously unselected package opendj.
(Reading database ... 185569 files and directories currently installed.)
Unpacking opendj (from opendj_3.5.3-1_all.deb) ...

Setting up opendj (3.5.3) ...
 Adding system startup for /etc/init.d/opendj ...
   /etc/rc0.d/K20opendj -> ../init.d/opendj
   /etc/rc1.d/K20opendj -> ../init.d/opendj
   /etc/rc6.d/K20opendj -> ../init.d/opendj
   /etc/rc2.d/S20opendj -> ../init.d/opendj
   /etc/rc3.d/S20opendj -> ../init.d/opendj
   /etc/rc4.d/S20opendj -> ../init.d/opendj
   /etc/rc5.d/S20opendj -> ../init.d/opendj

Processing triggers for ureadahead ...
ureadahead will be reprofiled on next reboot
```

The Debian package installs OpenDJ directory server in the `/opt/opendj` directory, generates service management scripts, adds documentation files under `/usr/share/doc/opendj`, and adds man pages under `/opt/opendj/share/man`.

The files are owned by root by default, making it easier to have OpenDJ listen on ports 389 and 636.

3. Configure OpenDJ directory server by using the command **sudo /opt/opendj/setup**:

```
$ sudo /opt/opendj/setup --cli
...
To see basic server configuration status and configuration you can launch
 /opt/opendj/bin/status
```

4. (Optional) Check OpenDJ directory server status:

```
$ service opendj status
$opendj status: > Running.
$ sudo /opt/opendj/bin/status


>>>> Specify OpenDJ LDAP connection parameters

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

          --- Server Status ---
Server Run Status:        Started
Open Connections:         1

          --- Server Details ---
Host Name:                ubuntu.example.com
Administrative Users:     cn=Directory Manager
```

```
Installation Path:        /opt/opendj
Version:                  OpenDJ 3.5.3
Java Version:             version
Administration Connector: Port 4444 (LDAPS)

          --- Connection Handlers ---
Address:Port : Protocol               : State
-------------:-----------------------:--------
-
--           : LDIF                   : Disabled
0.0.0.0:161  : SNMP                   : Disabled
0.0.0.0:389  : LDAP (allows StartTLS) : Enabled
0.0.0.0:636  : LDAPS                  : Enabled
0.0.0.0:1689 : JMX                    : Disabled
0.0.0.0:8080 : HTTP                   : Disabled

          --- Data Sources ---
Base DN:     dc=example,dc=com
Backend ID:  userRoot
Entries:     2002
Replication:
```

To Install From the RPM Package

On Red Hat and related Linux distributions such as Fedora and CentOS, you can install OpenDJ directory server from the RPM package:

1. Log in as superuser to install the software:

   ```
   $ su
   Password:
   #
   ```

2. Before you install OpenDJ, install a Java runtime environment if none is installed yet.

   You might need to download an RPM to install the Java runtime environment, and then install the RPM by using the **rpm** command:

   ```
   # rpm -ivh jre-*.rpm
   ```

3. Install the OpenDJ directory server package:

   ```
   # rpm -i opendj-3.5.3-1.noarch.rpm
   Pre Install - initial install
   Post Install - initial install

   #
   ```

The RPM package installs OpenDJ directory server in the `/opt/opendj` directory, generates service management scripts, and adds man pages under `/opt/opendj/share/man`.

The files are owned by root by default, making it easier to have OpenDJ listen on ports 389 and 636.

4. Configure OpenDJ directory server by using the command **/opt/opendj/setup**:

```
# /opt/opendj/setup --cli
...
To see basic server configuration status and configuration you can launch
 /opt/opendj/bin/status
```

5. (Optional) Check OpenDJ directory server status:

```
# service opendj status
opendj status: > Running.
# /opt/opendj/bin/status


>>>> Specify OpenDJ LDAP connection parameters

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

          --- Server Status ---
Server Run Status:        Started
Open Connections:         1

          --- Server Details ---
Host Name:                fedora.example.com
Administrative Users:     cn=Directory Manager
Installation Path:        /opt/opendj
Version:                  OpenDJ 3.5.3
Java Version:             version
Administration Connector: Port 4444 (LDAPS)

          --- Connection Handlers ---
Address:Port : Protocol               : State
-------------:-----------------------:--------
-
--           : LDIF                   : Disabled
0.0.0.0:161  : SNMP                   : Disabled
0.0.0.0:389  : LDAP (allows StartTLS) : Enabled
0.0.0.0:636  : LDAPS                  : Enabled
0.0.0.0:1689 : JMX                    : Disabled
0.0.0.0:8080 : HTTP                   : Disabled

          --- Data Sources ---
Base DN:     dc=example,dc=com
Backend ID:  userRoot
Entries:     2002
Replication:
```

By default OpenDJ starts in run levels 2, 3, 4, and 5:

```
# chkconfig --list | grep opendj
...
opendj         0:off    1:off    2:on    3:on    4:on    5:on    6:off
```

To Install OpenDJ Directory Server With a Properties File

You can install OpenDJ directory server by using the **setup** command with a properties file.

Property names correspond to the option names, but without leading dashes. Options that take no arguments become boolean properties as in the following example:

```
enableStartTLS=true
```

If you use a properties file with multiple tools, prefix the property name with the tool name followed by a dot (`.`), in the following example:

```
setup.rootUserPasswordFile=/tmp/pwd.txt
```

The following steps demonstrate use of a properties file as part of a scripted installation process:

1. Prepare your properties file.

   This procedure uses the following example properties file:

   ```
   #
   # Sample properties file to set up OpenDJ directory server
   #
   hostname                        =opendj.example.com
   ldapPort                        =1389
   generateSelfSignedCertificate   =true
   enableStartTLS                  =true
   ldapsPort                       =1636
   jmxPort                         =1689
   adminConnectorPort              =4444
   rootUserDN                      =cn=Directory Manager
   rootUserPassword                =password
   baseDN                          =dc=example,dc=com
   ldifFile                        =/net/install/dj/Example.ldif
   #sampleData                     =2000
   ```

   If you have multiple servers to install, consider scripting creation of the properties files.

2. Prepare an installation script:

```
$ cat /net/install/dj/1/setup.sh
#!/bin/sh

unzip -d /path/to /net/install/dj/opendj-3.5.3.zip && cd /path/to/
opendj
./setup --cli --propertiesFilePath /net/install/dj/1/setup.props \
  --acceptLicense --no-prompt
```

The properties file contains only installation options, and does not fully configure OpenDJ directory server.

If you also want your script to configure OpenDJ directory server, follow a successful run of the **setup** command with **dsconfig** commands to configure the server. To run a series of configuration commands as a batch using the **dsconfig** command, use either the `--batchFilePath` *`file`* option, where *file* contains the configuration commands, or the `--batch` option to read from standard input as in the following example that creates a backend and sets up indexes:

```
/path/to/opendj/bin/dsconfig \
 --port 4444 \
 --hostname opendj.example.com \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --no-prompt \
 --trustAll \
 --batch <<END_OF_COMMAND_INPUT
 create-backend         --backend-name newBackend \
                        --type pdb \
                        --set base-dn:"dc=example,dc=org" \
                        --set db-cache-percent:20 \
                        --set enabled:true
 create-backend-index   --backend-name newBackend \
                        --type generic \
                        --set index-type:equality \
                        --set index-type:substring \
                        --index-name cn
 create-backend-index   --backend-name newBackend \
                        --type generic \
                        --set index-type:equality \
                        --set index-type:substring \
                        --index-name sn
 create-backend-index   --backend-name newBackend \
                        --type generic \
                        --set index-type:equality \
                        --index-name uid
 create-backend-index   --backend-name newBackend \
                        --type generic \
                        --set index-type:equality \
                        --set index-type:substring \
                        --index-name mail
END_OF_COMMAND_INPUT
```

3. Run your installation script:

```
$ /net/install/dj/1/setup.sh
Archive:  /net/install/dj/opendj-3.5.3.zip
   creating: /path/to/
opendj
...
  inflating: /path/to/opendj/setup
  inflating: /path/to/opendj/uninstall
  inflating: /path/to/opendj/upgrade

READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING
THE FORGEROCK SOFTWARE, YOU, ON BEHALF OF YOURSELF AND YOUR COMPANY, AGREE TO
BE BOUND BY THIS SOFTWARE LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THESE
TERMS, DO NOT DOWNLOAD OR INSTALL THE FORGEROCK SOFTWARE.

...

Do you accept the License Agreement?yes
See /var/folders/.../opendj-setup-....log for a detailed log of this operation.

Configuring Directory Server ..... Done.
Configuring Certificates ..... Done.
Importing LDIF file /net/install/dj/Example.ldif ....... Done.
Starting Directory Server ....... Done.

To see basic server configuration status and configuration you can launch
 /path/to/opendj/bin/status
```

At this point you can use OpenDJ directory server, or you can perform additional configuration.

To Move Data from a PDB Backend to a JE Backend

Although the **dsconfig** command does not provide a way to change a database backend type, you can move data from a PDB Backend to a JE Backend as demonstrated by the script shown in "Example Script for Changing a PDB Backend to a JE Backend". Alternatively, follow these steps:

1. List the indexes configured for the PDB backend.

   The following example shows indexes for a `userRoot` PDB backend:

```
$ dsconfig \
 list-backend-indexes \
 --port 4444 \
 --hostname opendj.example.com \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --backend-name userRoot \
 --no-prompt \
 --trustAll
Backend Index    : index-type          : index-entry-limit : index-extensible-matching-rule :
 confidentiality-
enabled
-----------------:---------------------:-------------------:--------------------------------:---------
---------------
aci              : presence            : 4000              : -                              : false
cn               : equality, substring : 4000              : -                              : false
ds-sync-conflict : equality            : 4000              : -                              : false
ds-sync-hist     : ordering            : 4000              : -                              : false
entryUUID        : equality            : 4000              : -                              : false
givenName        : equality, substring : 4000              : -                              : false
mail             : equality, substring : 4000              : -                              : false
member           : equality            : 4000              : -                              : false
objectClass      : equality            : 4000              : -                              : false
sn               : equality, substring : 4000              : -                              : false
telephoneNumber  : equality, substring : 4000              : -                              : false
uid              : equality            : 4000              : -                              : false
uniqueMember     : equality            : 4000              : -                              : false
```

2. Export the data in the PDB backend to LDIF.

   For instructions, see "Importing and Exporting Data" in the *Administration Guide*.

3. Delete the PDB backend.

   For instructions, see "Deleting a Database Backend" in the *Administration Guide*.

4. Create a JE backend.

   For instructions, see "Creating a New Database Backend" in the *Administration Guide*.

5. Create the same indexes for the JE backend that were present in the PDB backend.

   For instructions, see "Configuring and Rebuilding Indexes" in the *Administration Guide*.

6. Import the data from LDIF into the JE backend.

Example Script for Changing a PDB Backend to a JE Backend

The following Bash script demonstrates how to change a PDB backend to a JE Backend:

```
#!/usr/bin/env bash
#
# The contents of this file are subject to the terms of the Common Development and
# Distribution License (the License). You may not use this file except in compliance with the
# License.
#
```

```
# You can obtain a copy of the License at legal-notices/CDDLv1.0.txt. See the License for the
# specific language governing permission and limitations under the License.
#
# When distributing Covered Software, include this CDDL Header Notice in each file and include
# the License file at legal-notices/CDDLv1.0.txt. If applicable, add the following below the CDDL
# Header, with the fields enclosed by brackets [] replaced by your own identifying
# information: "Portions Copyright [year] [name of copyright owner]".
#
# Copyright 2017-2018 ForgeRock AS.
#

if test $# -ne 1
then
  echo "Usage: $0 backendID"
  echo "Migrate a PDB backend to a JE backend with all the data."
  echo "Run this script from the server base directory, such as /path/to/opendj."
  exit 1
fi

# Check that the server is stopped.
echo "Verifying that the server is stopped..."
./bin/status -n -s > /dev/null
if test $? -ne 0
then
  echo "The Directory Server must be stopped to migrate a backend."
  echo "Please stop the server and relaunch the script."
  exit 1
fi
echo ""

# Check for instance.loc.
LOC=.
if [ -f ./instance.loc ]
then
  LOC=`cat ./instance.loc`
elif [ -f /etc/opendj/instance.loc ]
then
  LOC=`cat /etc/opendj/instance.loc`
fi

# Check the backendID.
echo "Verifying the backend $1"
DN=`./bin/ldifsearch --ldifFile "$LOC"/config/config.ldif "(&(objectclass=ds-cfg-pdb-backend)(ds-cfg-
backend-id=$1))" dn | grep "^dn:"`
if [ -z "$DN" ]
then
  echo "Could not find a PDB backend with this name. Exiting."
  exit 2
fi

echo "Exporting data to /tmp/data_$$"
# Export data from the PDB backend.
./bin/export-ldif -n "$1" -l /tmp/data_$$
if test $? -ne 0
then
  echo "Export from PDB failed."
  exit 3
fi
```

```
echo "Updating configuration"
# Change the PDB backend configuration to a JE backend configuration.
cat > /tmp/changes_$$ << EOF
$DN
changetype: modify
delete: objectClass
objectClass: ds-cfg-pdb-backend
-
add: objectClass
objectClass: ds-cfg-je-backend
-
replace: ds-cfg-java-class
ds-cfg-java-class: org.opends.server.backends.jeb.JEBackend
EOF

./bin/ldifmodify --targetLDIF "$LOC"/config/config.ldif.$$ --sourceLDIF "$LOC"/config/config.ldif --
changesLDIF /tmp/changes_$$
if test $? -ne 0
then
  echo "Modifications failed. Restoring the original configuration"
  rm /tmp/changes_$$
  exit 4
fi

cp "$LOC"/config/config.ldif.$$ "$LOC"/config/config.ldif
echo "Configuration updates done."
echo "Importing data..."
# Import the data into the JE backend.
./bin/import-ldif -n $1 -l /tmp/data_$$
if test $? -ne 0
then
  echo "Importing data failed."
  echo "The exported data file is /tmp/data_$$"
  exit 5
fi
echo "Backend $1 converted successfully from PDB to JE."
rm /tmp/data_$$
rm /tmp/changes_$$
rm "$LOC"/config/config.ldif.$$
```

To Install OpenDJ REST to LDAP Gateway

The OpenDJ REST to LDAP gateway functions as a web application in a web application container, running independently of OpenDJ. Alternatively, you can use the HTTP connection handler in OpenDJ directory server. For instructions see "To Set Up REST Access to User Data" in the *Administration Guide*.

You configure the gateway to access your directory service by editing configuration files in the deployed web application:

`WEB-INF/classes/config.json`

This file defines how the gateway connects to LDAP directory servers, and how user identities extracted from HTTP requests map to LDAP user identities.

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

`WEB-INF/classes/logging.properties`

This file defines logging properties, and can be used when the gateway runs in Apache Tomcat.

`WEB-INF/classes/rest2ldap/rest2ldap.json`

This file defines which LDAP features the gateway uses.

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

`WEB-INF/classes/rest2ldap/endpoints/api/example-v1.json`

This file defines JSON resource to LDAP entry mappings.

You can edit this file, and define additional files for alternative APIs and versions of APIs. For details, see "Mapping Configuration File" in the *Reference*.

Follow these steps to install the OpenDJ REST to LDAP gateway:

1. Deploy `opendj-rest2ldap-servlet-3.5.3.war` according to the instructions for your application server.
2. Edit the configuration files in the deployed gateway web application.

   At minimum adjust the following configuration settings in `WEB-INF/classes/config.json`:

   - `primaryLDAPServers`: Set to the correct directory server host names and port numbers.
   - `authentication`: Set to the correct simple bind credentials.

     The LDAP account used to authenticate needs to perform proxied authorization as described in "Configuring Proxied Authorization" in the *Directory Server Developer's Guide*.

   The default sample configuration configuration is built to work with generated example data and also the sample content in Example.ldif. If your data is different, then you must also change the JSON resource to LDAP entry mapping settings, described in "Mapping Configuration File" in the *Reference*.

   For details regarding the configuration, see "*REST to LDAP Configuration*" in the *Reference*.

   When connecting to directory servers over LDAPS or LDAP and StartTLS, you can configure the trust manager to use a file-based truststore for server certificates that the gateway should trust. This allows the gateway to validate server certificates signed, for example, by a Certificate Authority not recognized by the Java environment when setting up LDAPS or StartTLS connections. See "Preparing For Secure Communications" in the *Administration Guide* for an example of how to use the Java **keytool** command to import a server certificate into a truststore file.
3. (Optional) If necessary, adjust the log level.

Log levels are defined in java.util.logging.Level.

By default, the log level is set to `INFO`, and the gateway logs HTTP request-related messages. To have the gateway log LDAP request-related messages, set the log level to `FINEST` in one of the following ways:

- If the REST to LDAP gateway runs in Apache Tomcat, edit `WEB-INF/classes/logging.properties` to set `org.forgerock.opendj.rest2ldap.level = FINEST`. For details on Tomcat's implementation of the logging API, see *Logging in Tomcat*.

  Messages are written to *`CATALINA_BASE`*`/logs/rest2ldap.`*`yyyy-MM-dd`*`.log`.

- If the REST to LDAP gateway runs in Jetty, make sure you set the log level system property when starting Jetty: `-Dorg.forgerock.opendj.rest2ldap.level=FINEST`.

  Messages are written to the Jetty log.

4. Restart the REST to LDAP gateway or the application server to make sure the configuration changes are taken into account.

5. Make sure that your directory server is running, and then check that the gateway is connecting correctly.

   The following command reads Babs Jensen's entry through the gateway to a directory server holding data from `Example.ldif`. In this example, the gateway is deployed under `/rest2ldap`:

```
$ curl http://bjensen:hifalutin@opendj.example.com:8080/rest2ldap/api/users/bjensen
{
  "_id" : "bjensen",
  "_rev" : "0000000084ebc394",
  "_schema" : "frapi:opendj:rest2ldap:posixUser:1.0",
  "_meta" : { },
  "userName" : "bjensen@example.com",
  "displayName" : [ "Barbara Jensen", "Babs Jensen" ],
  "name" : {
    "givenName" : "Barbara",
    "familyName" : "Jensen"
  },
  "description" : "Original description",
  "contactInformation" : {
    "telephoneNumber" : "+1 408 555 1862",
    "emailAddress" : "bjensen@example.com"
  },
  "uidNumber" : "1076",
  "gidNumber" : "1000",
  "homeDirectory" : "/home/bjensen",
  "manager" : {
    "_id" : "trigden",
    "displayName" : "Torrey Rigden"
  }
}
```

If you generated example data, Babs Jensen's entry is not included. Instead, try a URL such as `http://user.0:password@opendj.example.com:8080/rest2ldap/api/users/user.0`.

To Install OpenDJ REST to LDAP Gateway (3.0)

The OpenDJ REST to LDAP gateway functions as a web application in a web application container, running independently of OpenDJ. Alternatively, you can use the HTTP connection handler in OpenDJ directory server. For instructions see "To Set Up REST Access to OpenDJ Directory Server" in the *Administration Guide*.

Note

This procedure applies to OpenDJ REST to LDAP gateway 3.0. If you are using OpenDJ REST to LDAP gateway 3.5, see "To Install OpenDJ REST to LDAP Gateway".

You configure the gateway to access your directory service by editing the configuration file `opendj-rest2ldap-servlet.json` in the deployed OpenDJ REST to LDAP gateway web application:

1. Deploy `opendj-rest2ldap-servlet-3.5.3-servlet.war` according to the instructions for your application server.
2. Edit `opendj-rest2ldap-servlet.json` where you deployed the gateway web application.

   The default JSON resource for the configuration includes both connection and authentication information, and also `mappings`. The `mappings` describe how the gateway translates between JSON and LDAP representations of directory data. The default `mappings` are built to work with generated example data and also the sample content in Example.ldif.

   At minimum adjust the following gateway configuration settings:

   - `primaryLDAPServers`: Set to the correct directory server host names and port numbers
   - `authentication`: Set to the correct simple bind credentials
   - `mappings`: Make sure these match the directory data

   For details on the configuration see "*REST to LDAP Configuration*" in the *Reference*.

   When connecting to directory servers over LDAPS or LDAP and StartTLS, you can configure the trust manager to use a file-based truststore for server certificates that the gateway should trust. This allows the gateway to validate server certificates signed, for example, by a Certificate Authority not recognized by the Java environment when setting up LDAPS or StartTLS connections. See "Preparing For Secure Communications" in the *Administration Guide* for an example of how to use the Java **keytool** command to import a server certificate into a truststore file.
3. Restart the REST to LDAP gateway or the application server to make sure the configuration changes are taken into account.

4. Make sure that your directory server is running, and then check that the gateway is connecting correctly.

   The following command reads Babs Jensen's entry through the gateway to a directory server holding data from `Example.ldif`:

```
$ curl http://bjensen:hifalutin@opendj.example.com:8080/rest2ldap/users/bjensen
{
  "_rev" : "000000002ee3b764",
  "schemas" : [ "urn:scim:schemas:core:1.0" ],
  "contactInformation" : {
    "telephoneNumber" : "+1 408 555 1862",
    "emailAddress" : "bjensen@example.com"
  },
  "_id" : "bjensen",
  "name" : {
    "familyName" : "Jensen",
    "givenName" : "Barbara"
  },
  "userName" : "bjensen@example.com",
  "displayName" : "Barbara Jensen",
  "manager" : [ {
    "_id" : "trigden",
    "displayName" : "Torrey Rigden"
  } ]
}
```

   If you generated example data, Babs Jensen's entry is not included. Instead, try a URL such as `http://user.0:password@opendj.example.com:8080/rest2ldap/users/user.0`.

To Install OpenDJ DSML gateway

The OpenDJ DSML gateway functions as a web application in a web application container. The DSML gateway runs independently of OpenDJ directory server. You configure the gateway to access your directory service by editing the `ldap.host` and `ldap.port` parameters in the gateway `WEB-INF/web.xml` configuration file:

1. Deploy `opendj-dsml-servlet-3.5.3.war` according to the instructions for your application server.
2. Edit `WEB-INF/web.xml` to ensure the values for `ldap.host` and `ldap.port` are correct.
3. Restart the web application container according to the instructions for your application server.

Chapter 2
Upgrading to OpenDJ 3.5

This chapter covers upgrade from previous versions.

If the OpenDJ directory server version is older than 2.6.0, you must upgrade your deployment to use at least OpenDJ directory server 2.6.0 before following the procedures in this chapter. For details on upgrading to that version, see *Upgrading to OpenDJ 2.6.0*.

Tip

With the migration of OpenDJ project code from Subversion to Git, the upgrade code has changed to no longer rely on Subversion revision numbers.

As a result, upgrade from a nightly build is not guaranteed to work. Upgrade from one release to another works fine, as does upgrade from a release to a nightly build.

As a workaround, rather than upgrading from a nightly build, install a new server alongside the existing server and use replication to bring the new server up to date before retiring the older server.

This chapter includes the following procedures and examples:

Before You Upgrade

1. Prepare to perform the upgrade procedure as the user who owns the OpenDJ server files.

Make sure you have the credentials to run commands as the user who owns the server.

2. (Optional) If OpenDJ directory server runs with Java 6, move to a newer version before continuing the upgrade process.

   To move to a newer version, edit the `default.java-home` setting in the `opendj/config/java.properties` file, and then run the **dsjavaproperties** command.

3. (Optional) If you are upgrading to OpenDJ OEM edition from OpenDJ 2.6, make sure there is enough disk space to export all of the data to LDIF files.

4. Download enterprise software releases through the ForgeRock BackStage site. ForgeRock enterprise releases are thoroughly validated builds for ForgeRock customers who run OpenDJ in production deployments, and for those who want to try or test with release builds.

5. (Optional) If you are upgrading OpenDJ directory server on Windows, and OpenDJ is registered as a Windows service, disable OpenDJ as a Windows service before upgrade, as in the following example:

   ```
   C:\path\to\opendj\bat> windows-service.bat --disableService
   ```

   After upgrade, you can enable OpenDJ as a Windows service again.

6. Make sure you perform a full backup of your current OpenDJ installation to revert if the upgrade fails.

   Due to changes to the backup archive format, make sure you stop OpenDJ directory server and back up the file system directory where the current OpenDJ directory server is installed rather than creating a backup archive with the **backup** command.

To Upgrade to OpenDJ 3.5

If you are upgrading to the OEM edition from OpenDJ 2.6, then this procedure does not apply. Skip instead to "To Upgrade to OpenDJ OEM Edition".

Before starting this procedure, follow the steps in "Before You Upgrade".

To upgrade to OpenDJ directory server installed from native packages (.deb, .rpm), use the command-line package management tools provided by the system.

Note

OpenDJ directory server backend storage options have changed since OpenDJ 2.6. The underlying implementation is based on an extensible architecture, allowing you to choose the backend storage type when you create a persistent backend for directory data.

This procedure applies when you upgrade from OpenDJ 2.6, retaining the same underlying backend storage. The configuration changes from a Local DB backend to a JE Backend, and the upgrade procedure migrates the underlying backend database. There is no need to export data to LDIF when following this procedure.

The following steps describe how to upgrade OpenDJ directory server installed from the cross-platform (.zip) delivery:

1. Log in as the user who owns the current OpenDJ server.

2. Stop the current OpenDJ server.

3. (Optional) If you have not already backed up the current OpenDJ server, make a back up copy of the directory where OpenDJ is installed.

4. Unpack the new files from the .zip delivery over the current server files.

5. Run the **upgrade** command, described in upgrade(1) in the *Reference*, to bring OpenDJ configuration and application data up to date with the new binary and script files that you copied over the current server files.

   By default, the **upgrade** command requests confirmation before making important configuration changes. For some potentially long-duration tasks, such as rebuilding indexes, the default choice is to defer the tasks until after upgrade. Tasks that are not performed during upgrade must generally be performed after upgrade but before you restart the server.

   You can use the `--no-prompt` option to run the command non-interactively, with the `--acceptLicense` option to accept the license terms non-interactively.

   When using the `--no-prompt` option, if the **upgrade** command cannot complete because it requires confirmation for a potentially very long or critical task, then it exits with an error and a message about how to finish making the changes. You can add the `--force` option to force a non-interactive upgrade to continue in this case, also performing long running and critical tasks.

6. Start the upgraded OpenDJ server.

   At this point the upgrade process is complete. See the resulting `upgrade.log` file for a full list of operations performed.

   Note

   When you upgrade to OpenDJ 3.5 from an OpenDJ 3 or earlier, the upgrade procedure leaves the HTTP connection handler disabled.

   The newer configuration supports inheritance and subsresources, but is not compatible with the previous configuration.

You must rewrite your configuration to the version described in "*REST to LDAP Configuration*" in the *Reference*, and then reconfigure the server to use the new configuration. For details, see "RESTful Client Access Over HTTP" in the *Administration Guide*.

7. (Optional) If you are upgrading OpenDJ directory server on Windows, and you disabled OpenDJ as a Windows service in order to upgrade, enable OpenDJ as a Windows service again as in the following example:

```
C:\path\to\opendj\bat> windows-service.bat --enableService
```

Upgrading to OpenDJ 3.5

The following example upgrades an OpenDJ 2.6.3 directory server, backing up the current server directory in case the upgrade process fails. In this example, the server properties are updated to use Java 8, and the Local DB backend is migrated to a JE backend:

```
$ cd /path/to/
$ sed -e "s/default.java-home=.*/default.java-home=\/path\/to\/jdk1.8/" \
 opendj/config/java.properties \
 > opendj/config/java.properties.new ; \
 mv opendj/config/java.properties.new opendj/config/java.properties
$ /path/to/opendj/bin/dsjavaproperties
$ /path/to/opendj/bin/stop-ds --quiet
... msg=The Directory Server is now stopped
$ zip -rq OpenDJ-backup.zip opendj/
$ unzip -o ~/Downloads/opendj-3.5.3.zip
$ /path/to/opendj/upgrade --acceptLicense

>>>> OpenDJ Upgrade Utility

 * OpenDJ will be upgraded from version 2.6.3.12667 to
 3.5.3.build-hash
 * See '/path/to/opendj/upgrade.log' for a detailed log of this operation

>>>> Preparing to upgrade

  OpenDJ 3.5.3 introduced changes to the JE backend configuration and database
  format. The upgrade will update all JE backend configurations, but will only
  migrate JE backend databases which are associated with *enabled* JE
  backends. It is very strongly recommended that any existing data has been
  backed up and that you have read the upgrade documentation before
  proceeding. Do you want to proceed with the upgrade? (yes/no) [no]: yes

  OpenDJ 3.5.3 changed the matching rule implementations. All indexes have to
  be rebuilt. This could take a long time to proceed. Do you want to launch
  this process automatically at the end of the upgrade? (yes/no) [no]: yes

  OpenDJ 3.5.3 improved the replication changelog storage format. As a
  consequence, the old changelog content of the current replication server
  will be erased by the upgrade. The new changelog content will be
  automatically reconstructed from the changelog of other replication servers
```

```
  in the topology. After the upgrade, dsreplication reset-change-number can be
  used to reset the changelog change-number of the current replication server
  to match another replication server. Do you want to proceed with the
  upgrade? (yes/no) [no]: yes

  The upgrade is ready to proceed. Do you wish to continue? (yes/no) [yes]:


>>>> Performing upgrade

  Changing matching rule for 'userCertificate' and 'caCertificate' to
  CertificateExactMatch.............................................   100%
  Configuring 'CertificateExactMatch' matching rule.................   100%
  Replacing schema file '03-pwpolicyextension.ldif'.................   100%
  Removing 'dc=replicationchanges' backend..........................   100%
  Removing ACI for 'dc=replicationchanges'..........................   100%
  Adding default privilege 'changelog-read' to all root DNs.........   100%
  Adding PKCS5S2 password storage scheme configuration..............   100%
  Rerunning dsjavaproperties........................................   100%
  Updating ds-cfg-java-class attribute in File-Based Debug Logger.....   100%
  Deleting ds-cfg-default-debug-level attribute in File-Based Debug
  Logger............................................................   100%
  Updating ds-cfg-default-severity attribute in File-Based Error
  Logger............................................................   100%
  Updating ds-cfg-override-severity attribute in Replication Repair
  Logger............................................................   100%
  Removing config for 'Network Groups'..............................   100%
  Removing config for 'Workflows'...................................   100%
  Removing config for 'Workflow Elements'...........................   100%
  Removing config for 'Network Group Plugin'........................   100%
  Removing config for 'Extensions'..................................   100%
  Removing config for 'File System Entry Cache'.....................   100%
  Removing config for 'Entry Cache Preload'.........................   100%
  Removing file '/path/to/opendj/bin/dsframework'...................   100%
  Removing file '/path/to/opendj/bat/dsframework.bat'...............   100%
  Migrating JE backend 'userRoot'...................................   100%
  Convert local DB backends to JE backends..........................   100%
  Convert local DB indexes to backend indexes.......................   100%
  Convert local DB VLV indexes to backend VLV indexes...............   100%
  Removing file '/path/to/opendj/bin/dbtest'........................   100%
  Removing file '/path/to/opendj/bat/dbtest.bat'....................   100%
  Removing content of changelog in '/path/to/opendj/./changelogDb'
  directory.........................................................   100%
  Enable log file based replication changelog storage...............   100%
  Replacing schema file '02-config.ldif'............................   100%
  Archiving concatenated schema.....................................   100%

>>>> OpenDJ was successfully upgraded from version 2.6.3.12667 to
3.5.3.build-hash


>>>> Performing post upgrade tasks

...

>>>> Post upgrade tasks complete

 * See '/path/to/opendj/upgrade.log' for a detailed log of this operation
```

```
$ /path/to/opendj/bin/start-ds --quiet
$
```

To Upgrade to OpenDJ OEM Edition

If you are not upgrading to the OEM edition from OpenDJ 2.6, then this procedure does not apply. Skip instead to "To Upgrade to OpenDJ 3.5".

Before starting this procedure, follow the steps in "Before You Upgrade".

Note

OpenDJ directory server backend storage options have changed since OpenDJ 2.6. The underlying implementation is based on an extensible architecture, allowing you to choose the backend storage type when you create a persistent backend for directory data.

This procedure applies when you upgrade to the OEM edition from OpenDJ 2.6, changing the underlying backend storage. The configuration changes from a Local DB backend to a PDB Backend, but the **upgrade** command in this version *deletes the data from OpenDJ directory server*. Follow the instructions in this procedure to avoid data loss.

Follow these steps:

1. Login as the user who owns the current OpenDJ server.
2. Stop the current OpenDJ server.
3. Export all of the data to LDIF files.

   OpenDJ directory server OEM edition uses a new backend type, PDB. This edition does not support the older Local DB backend type. The upgrade process transforms the configuration to use the new backend type, but it does not export and import directory data. You must export the data, unpack the files of the new version over the old, run the upgrade, and then import the data.

   The following example exports Example.com data from the `userRoot` backend to an LDIF file:

   ```
   $ export-ldif --backendID userRoot --ldifFile ../ldif/Example.ldif
   ```

4. If you have not already backed up the current OpenDJ server, make a back up copy of the directory where OpenDJ is installed.
5. Unpack the new files over the current server files:

   - When upgrading the .zip distribution, overwrite the current files.

     The following example overwrites the current files with the new files:

     ```
     $ cd /path/to ; unzip -o ~/Downloads/opendj-3.5.3.zip
     ```

- When upgrading native packaging, use the command-line package management tools provided by the system to remove the 2.6 package, and then install the new package.

  For details, see "To Uninstall the Debian Package" or "To Uninstall the RPM Package", and "To Install From the Debian Package" or "To Install From the RPM Package".

6. Run the **upgrade** command to bring OpenDJ configuration and schema data up to date with the new binary and script files that replaced existing server files.

   By default, the **upgrade** command requests confirmation before making important configuration changes. For some potentially long-duration tasks, such as rebuilding indexes, the default choice is to defer the tasks until after upgrade. Tasks that are not performed during upgrade must generally be performed after upgrade but before you restart the server.

   You can use the `--no-prompt` option to run the command non-interactively, with the `--acceptLicense` option to accept the license terms non-interactively.

   When using the `--no-prompt` option, if the **upgrade** command cannot complete because it requires confirmation for a potentially very long or critical task, then it exits with an error and a message about how to finish making the changes. You can add the `--force` option to force a non-interactive upgrade to continue in this case, also performing long running and critical tasks.

   Once this step is complete, OpenDJ directory server no longer has access to user data that was stored in Local DB backends.

7. (Optional) If user data occupies significant disk space, and not enough disk space is available, then remove binary backups of the user data that you exported to LDIF.

   The upgrade process moves old user backend data to `opendj/db/*.bak` directories. This old user backend data is not accessible after upgrade. You can remove the old user backend data as shown in the following example:

   ```
   $ rm -rf /path/to/opendj/db/*.bak
   ```

8. Import all of the data from LDIF files.

   The following example imports Example.com data from an LDIF file to the `userRoot` backend:

   ```
   $ cd opendj/bin ; import-ldif --backendID userRoot --ldifFile ../ldif/Example.ldif
   ```

   Make sure you perform this step *for all user data backends*.

9. Start the upgraded OpenDJ server.

   Replication updates the upgraded server with changes that occurred during the upgrade process.

At this point the upgrade process is complete. See the resulting `upgrade.log` file for a full list of operations performed.

Upgrading To OpenDJ OEM Edition

The following example upgrades an OpenDJ 2.6.3 directory server to OpenDJ OEM edition, where the backend type for data storage is PDB. With the OEM edition, Local DB and JE backends are not supported. In this example, the server properties are updated to use Java 8, and the Local DB backend configuration is converted to use PDB backend. The directory data is exported to LDIF before upgrade, and imported from LDIF after upgrade:

```
$ cd /path/to/
$ sed -e "s/default.java-home=.*/default.java-home=\/path\/to\/jdk1.8/" \
 opendj/config/java.properties \
 > opendj/config/java.properties.new ; \
 mv opendj/config/java.properties.new opendj/config/java.properties
$ /path/to/opendj/bin/dsjavaproperties
$ /path/to/opendj/bin/stop-ds --quiet
... msg=The Directory Server is now stopped
$ /path/to/opendj/bin/export-ldif --backendID userRoot \
 --ldifFile opendj/ldif/Example.ldif
$ zip -rq opendj-backup.zip opendj/
$ unzip -o ~/Downloads/opendj-oem-3.5.3.zip
$ /path/to/opendj/upgrade --acceptLicense

>>>> OpenDJ Upgrade Utility

 * OpenDJ will be upgraded from version 2.6.3.12667 to
 3.5.3.build-hash
 * See '/path/to/opendj/upgrade.log' for a detailed log of this operation

>>>> Preparing to upgrade

  WARNING: OpenDJ 3.5.3 OEM Edition removes support for the Berkeley JE
  backend.

  The upgrade tool will reconfigure all JE backends as PDB backends.

  After the upgrade the new PDB backend(s) will be empty. It is therefore very
  strongly recommended that any data that was in the JE backends be exported
  to LDIF so that it can be re-imported once the upgrade completes.

  Do you want to make this configuration change? (yes/no) [no]: yes

  OpenDJ 3.5.3 changed the matching rule implementations. All indexes have to
  be rebuilt. This could take a long time to proceed. Do you want to launch
  this process automatically at the end of the upgrade? (yes/no) [no]: yes

  OpenDJ 3.5.3 improved the replication changelog storage format. As a
  consequence, the old changelog content of the current replication server
  will be erased by the upgrade. The new changelog content will be
  automatically reconstructed from the changelog of other replication servers
  in the topology. After the upgrade, dsreplication reset-change-number can be
  used to reset the changelog change-number of the current replication server
  to match another replication server. Do you want to proceed with the
```

```
  upgrade? (yes/no) [no]: yes

  The upgrade is ready to proceed. Do you wish to continue? (yes/no) [yes]:


>>>> Performing upgrade

  Changing matching rule for 'userCertificate' and 'caCertificate' to
  CertificateExactMatch.....................................................   100%
  Configuring 'CertificateExactMatch' matching rule.........................   100%
  Replacing schema file '03-pwpolicyextension.ldif'.........................   100%
  Removing 'dc=replicationchanges' backend..................................   100%
  Removing ACI for 'dc=replicationchanges'..................................   100%
  Adding default privilege 'changelog-read' to all root DNs.................   100%
  Adding PKCS5S2 password storage scheme configuration......................   100%
  Rerunning dsjavaproperties................................................   100%
  Updating ds-cfg-java-class attribute in File-Based Debug Logger.....       100%
  Deleting ds-cfg-default-debug-level attribute in File-Based Debug
  Logger....................................................................   100%
  Updating ds-cfg-default-severity attribute in File-Based Error
  Logger....................................................................   100%
  Updating ds-cfg-override-severity attribute in Replication Repair
  Logger....................................................................   100%
  Removing config for 'Network Groups'......................................   100%
  Removing config for 'Workflows'...........................................   100%
  Removing config for 'Workflow Elements'...................................   100%
  Removing config for 'Network Group Plugin'................................   100%
  Removing config for 'Extensions'..........................................   100%
  Removing config for 'File System Entry Cache'.............................   100%
  Removing config for 'Entry Cache Preload'.................................   100%
  Removing file '/path/to/opendj/bin/dsframework'...........................   100%
  Removing file '/path/to/opendj/bat/dsframework.bat'.......................   100%
  Removing file '/path/to/opendj/lib/je.jar'................................   100%
  Renaming local-db backend directory '/path/to/opendj/db/userRoot'
  to '/path/to/opendj/db/userRoot.bak'......................................   100%
  Reconfiguring local-db backends to PDB backends...........................   100%
  Reconfiguring local-db backend indexes to PDB backend indexes.....       100%
  Reconfiguring local-db backend VLV indexes to PDB backend VLV
  indexes...................................................................   100%
  Removing file '/path/to/opendj/bin/dbtest'................................   100%
  Removing file '/path/to/opendj/bat/dbtest.bat'............................   100%
  Removing content of changelog in '/path/to/opendj/./changelogDb'
  directory.................................................................   100%
  Enable log file based replication changelog storage.......................   100%
  Replacing schema file '02-config.ldif'....................................   100%
  Archiving concatenated schema.............................................   100%

>>>> OpenDJ was successfully upgraded from version 2.6.3.12667 to
3.5.3.build-hash


>>>> Performing post upgrade tasks

  [!] You must reimport all your data into the PDB backends in order to have a
  fully functional server
  ...

>>>> Post upgrade tasks complete
```

```
 * See '/path/to/opendj/upgrade.log' for a detailed log of this operation

$ /path/to/opendj/bin/import-ldif --backendID userRoot \
 --ldifFile opendj/ldif/Example.ldif
$ /path/to/opendj/bin/start-ds --quiet
# Optionally remove Local DB backup data:
$ rm -rf /path/to/opendj/db/userRoot.bak/
```

To Upgrade Replicated Servers

Important

The OpenDJ directory server upgrade process is designed to support a rolling (sequential) upgrade of replicated servers.

Do not upgrade all replicated servers at once in parallel, as this removes all replication changelog data simultaneously, breaking replication.

For each server in the replication topology, follow these steps:

1. Direct client application traffic away from the server to upgrade.
2. Upgrade the server as described above.
3. Direct client application traffic back to the upgraded server.

To Add a New Replica to an Existing Topology

Newer OpenDJ servers have updates to LDAP schema that enable support for some new features. The newer schemas are not all compatible with older servers.

When adding a new server to a replication topology with older servers and following the instructions in "Enabling Replication" in the *Administration Guide*, also follow these recommendations:

1. Enable replication using the **dsreplication** command delivered with the new server.
2. Use the `--noSchemaReplication` or the `--useSecondServerAsSchemaSource` option to avoid copying the newer schema to the older server.

   It is acceptable to copy the older schema to the newer server, though it prevents use of new features that depend on newer schema.
3. If some applications depend on Internet-Draft change numbers, see "To Align Draft Change Numbers" in the *Administration Guide*.

To Upgrade OpenDJ REST to LDAP Gateway

1. Rewrite your configuration to work with the new formats described in "*REST to LDAP Configuration*" in the *Reference*.

2. Replace the gateway web application with the newer version, as for a fresh installation.

To Upgrade OpenDJ DSML Gateway

- Replace the gateway web application with the newer version, as for a fresh installation.

Chapter 3
Removing OpenDJ Servers

This chapter includes the following procedures:

To Remove OpenDJ With the GUI Uninstaller

1. Run the **uninstall** command, described in uninstall(1) in the *Reference*.

   (UNIX) Run **/path/to/opendj/uninstall**.

   (Windows) Double-click `/path/to/opendj\uninstall.bat`.

   (Mac OS X) Double-click `/path/to/opendj/Uninstall.app`.

   The Uninstall Options screen appears.

2. Select the components to remove in the Uninstall Options screen, and then click Uninstall to proceed.

3. To complete the process, manually remove any remaining components indicated in the Finished screen.

To Uninstall OpenDJ From the Command-Line

1. Login as the user who installed and runs the server.

2. Run the **/path/to/opendj/uninstall --cli** command.

   This command starts the removal program in interactive mode on the command-line, prompting you for each option. Alternatively, use additional **uninstall** options to specify choices for the options. See **uninstall --help** for more information:

```
$ /path/to/opendj/uninstall --cli
Do you want to remove all components of the server or select the components to
remove?

    1)  Remove all components
    2)  Select the components to be removed

    q)  quit

Enter choice [1]:

The server is currently running and must be stopped before uninstallation can
continue.
Stop the Server and permanently delete the files? (yes / no) [yes]:

Stopping Directory Server ..... Done.
Deleting Files under the Installation Path ..... Done.

The Uninstall Completed Successfully.
To complete the uninstallation, you must delete manually the following files
and directories:
/path/to/opendj/lib
See /var/....log for a detailed log of this operation.
```

3. If the command output tells you to delete files manually, then remove those remaining files to complete the process:

```
$ rm -rf /path/to/opendj
```

To Uninstall the Debian Package

When you uninstall the Debian package from the command-line, OpenDJ directory server is stopped if it is running:

- Remove the package from your system:

```
$ sudo dpkg -r opendj
(Reading database ... 185725 files and directories currently installed.)
Removing opendj ...
*Stopping OpenDJ server...
Stopping Server...
[03/Jun/2013:10:00:49 +0200] category=BACKEND severity=NOTICE
 msgID=9896306 msg=The backend userRoot is now taken offline
[03/Jun/2013:10:00:49 +0200] category=CORE severity=NOTICE
 msgID=458955 msg=The Directory Server is now stopped

*OpenDJ successfully removed

$
```

Removing the package does not remove your data or configuration. You must remove `/opt/opendj` manually to get rid of all files.

To Uninstall the RPM Package

When you uninstall the RPM package from the command-line, OpenDJ directory server is stopped if it is running.

- Remove the package from your system:

```
# rpm -e opendj
Pre Uninstall - uninstall
Stopping Server...
[03/Jun/2013:10:42:46 +0200] category=BACKEND severity=NOTICE
 msgID=9896306 msg=The backend userRoot is now taken offline
[03/Jun/2013:10:42:46 +0200] category=CORE severity=NOTICE
 msgID=458955 msg=The Directory Server is now stopped
Post Uninstall - uninstall
OpenDJ successfully removed.
#
```

Removing the package does not remove your data or configuration. You must remove `/opt/opendj` manually to get rid of all files.

Index