

Installation

This section goes through the required installation steps to get a production ready Snow Owl up and running on your machines.

Installation requirements

Client-side requirements

Hardware requirements

Memory	4 GB
Disk	1 GB free space
Operations System	64-bit Microsoft Windows 7, 8, 8.1, 10

Server-side requirements

Hardware requirements

CPU	20 Core Server (eg. Dual Intel Xeon E5 2650 V3)
Memory	96 GB
Disk	4 x 256 GB Primary SSD Drive RAID 10
Network	1 Gbps Dedicated Port

Software Requirements

	Supported Platforms	Supported Version(s)	Notes
Operating systems	Linux	CentOS (RHEL) 6.8 or Ubuntu 14.04 LTS	We recommend starting with a minimal install and adding packages later, as required. B2i Healthcare only officially supports Snow Owl running on 64-bit derivatives of x86 hardware.
Java	Oracle JDK	1.8.0 update 121	An installable archive can be downloaded from the JDK8 download page . Select the “Linux x64” edition.

	Supported Platforms	Supported Version(s)	Notes
Database	MySQL	5.7	Terminology contents are persisted using a MySQL database, downloadable from MySQL's yum repository
LDAP	OpenLDAP	2.4.x	Authentication and authorization of browsers, terminology editors, reviewers and administrators is performed through an LDAP server. Browsing and managing OpenLDAP instances can be done through the Apache Directory Studio application. We recommend installing the latest release from the corresponding Download Versions page on Apache's website.

Installing

In this section we'll run you through installing Snow Owl in a production environment on a CentOS 6.8 machine with an external MySQL database and LDAP based authentication/authorization.

Operating system

Install Red Hat Enterprise Linux or CentOS using the minimal ISO image. As hardware configurations and the corresponding exact installation steps can be different from machine to machine, please refer to the [installation guide](#) on Red Hat's site for details (covers both distributions).

NOTE

The text-based installer does not offer all options compared to the graphical one; you may have to connect a physical monitor, or use the built-in KVM management capabilities of the server (if supported) to perform the installation from the graphical environment. The installed system only needs an SSH connection for administration.

When creating the partition layout, keep in mind that Snow Owl Server requires at least 50-150 GB of disk space when branched terminology editing is used extensively.

After logging in to the installed system, update installed packages to the latest version and add

EPEL as a package repository for dependencies. For non-CentOS installations, please see the [usage instructions](#) on the EPEL wiki.

```
# yum update
# reboot
# yum install epel-release ①
```

① Works only if CentOS was installed

Create a non-login user for Snow Owl Server to run as:

```
# useradd -r -s /sbin/nologin snowowl
```

For optimal indexing performance, consult the "Production Deployment" sections of [Elasticsearch: The Definitive Guide](#). In particular, the following settings are applicable to an installation of the terminology server:

```
# sysctl settings, to be added to /etc/sysctl.conf or equivalent

vm.swappiness = 1
vm.max_map_count = 262144

# "noop" I/O scheduler, should be set in eg. /etc/rc.local for solid state disks:

echo noop > /sys/block/sda/queue/scheduler
```

Network

Install `system-config-firewall-tui`:

```
# yum install dbus dbus-python system-config-firewall-tui
# reboot
```

Using the text-based UI, enable these Trusted Services:

SSH

For remote administration of the server

Also open access to the following ports:

8080/TCP

Used by Snow Owl Server's REST API

2036/TCP

Used by the Net4J binary protocol connecting Snow Owl clients to the server

Database

An extensive installation guide for getting MySQL Community Edition from a yum repository is available at [the MySQL Documentation Library](#). The required steps are summarized below.

Install MySQL's yum repository with the following command:

```
# yum install -y mysql57-community-release
```

The repository for MySQL 5.7 should be enabled by default. Confirm by opening `/etc/yum.repos.d/mysql-community.repo`:

```
# yum repolist enabled | grep mysql ①
```

① 5.7 should appear somewhere

Install MySQL Community Server using yum (also run an update so packages get replaced with the community version):

```
# yum install -y mysql-community-server
# yum update -y
```

Start the service and wait for first-time initialization to complete:

```
# chkconfig --list mysqld
# service mysqld start
Starting MySQL. [ OK ]
```

After a few minutes, check if the database service is still running and enabled at startup:

```
# service mysqld status
mysqld (pid 1757) is running...
```

```
# chkconfig mysqld on
# chkconfig --list mysqld
mysqld          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

Get the temporary password and change it

```
# grep 'temporary password' /var/log/mysqld.log
# mysql -uroot -p
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

NOTE

The entered new root password will be used later for configuration and administrative purposes; do not forget this password.

Edit `/etc/my.cnf` to adjust settings for the MySQL server. Recommended settings are shown below, but there are lots of additional tunable settings to choose from depending on the hardware configuration used; please see <https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html> for the full set of system variables.

/etc/my.cnf

```
#
# The following options will be read by MySQL client applications.
# Note that only client applications shipped by MySQL are guaranteed
# to read this section. If you want your own MySQL client program to
# honor these values, you need to specify it as an option during the
# MySQL client library initialization.
#
[client]
socket = /var/lib/mysql/mysql.sock

[mysqld]

# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
innodb_buffer_pool_size = 5G

# Size of each log file in a log group. You should set the combined size
# of log files to about 25%-100% of your buffer pool size to avoid
# unneeded buffer pool flush activity on log file overwrite. However,
# note that a larger logfile size will increase the time needed for the
# recovery process.
innodb_log_file_size = 1G

# Total number of files in the log group. A value of 2-3 is usually good
# enough.
innodb_log_files_in_group = 3

# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin

# These are commonly set, remove the # and set as required.
# basedir = .....
# datadir = .....
# port = .....
# server_id = .....
socket = /var/lib/mysql/mysql.sock

# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
```

```
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M

sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES

# Minimum word length to be indexed by the full text search index.
# You might wish to decrease it if you need to search for shorter words.
# Note that you need to rebuild your FULLTEXT index, after you have
# modified this value.
ft_min_word_len = 2

# The maximum size of a query packet the server can handle as well as
# maximum query size server can process (Important when working with
# large BLOBs).  enlarged dynamically, for each connection.
max_allowed_packet = 16M

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

collation-server=utf8_unicode_ci
character-set-server=utf8

lower_case_table_names=1
transaction-isolation=READ-COMMITTED

[mysqldump]
# Do not buffer the whole result set in memory before writing it to
# file. Required for dumping very large tables
quick
max_allowed_packet = 16M
```

Restart the mysql service to make sure changes are picked up:

```
# service mysqld restart
Stopping mysqld:           [ OK ]
Starting mysqld:           [ OK ]
```

Configuring database

Create a MySQL user for the Snow Owl Server by connecting to the DBMS via the console:

```
$ mysql -u root -p
Enter password: root_pwd ①

mysql> CREATE USER 'snowowl'@'localhost' IDENTIFIED BY 'snowowl_pwd'; ②
```

- ① Replace `root_pwd` with the password for the `root` user in MySQL
- ② Replace `snowowl_pwd` with a generated password for the `snowowl` user in MySQL

Save the following shell script to an executable file to create databases and grant privileges for user `snowowl`:

`snowowl_create_db.sh`

```
#!/usr/bin/env bash

#
# Copyright 2015-2017 B2i Healthcare Pte Ltd, http://b2i.sg
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#
# This script creates all required databases and user for Snow Owl Server US edition.
#

DATABASES=( snomedStore )

MYSQL='which mysql'
USER="root"
PASSWORD="root_pwd"

echo -e "\nStarting Snow Owl database setup procedure."

# Create user
${MYSQL} -u${USER} -p${PASSWORD} -e "CREATE USER 'snowowl'@'localhost' identified by
'snowowl';" > /dev/null 2>&1
echo -e "\n\tCreated snowowl/mysql user."

# Create databases
echo -e "\n\tCreating Snow Owl databases:"
for i in "${DATABASES[@]}"
do
    ${MYSQL} -u${USER} -p${PASSWORD} -e "CREATE DATABASE \"${i}\" DEFAULT CHARSET
'utf8';" > /dev/null 2>&1
    echo -e "\t\tCreated Snow Owl database ${i}."
    ${MYSQL} -u${USER} -p${PASSWORD} -e "GRANT ALL PRIVILEGES ON \"${i}\".* to
'snowowl'@'localhost';" > /dev/null 2>&1
```



```
        echo -e "\t\tPrivileges granted on ${i} for snowowl user."
    done
    echo -e "\tCreation of Snow Owl databases are now complete."

    ${MYSQL} -u${USER} -p${PASSWORD} -e "FLUSH PRIVILEGES;" > /dev/null 2>&1
    echo -e "\n\tGrant tables reloaded."

    echo -e "\nSnow Owl database setup procedure has finished.\n"
```

B2i provided MySQL dumps (if present) can be found in `/opt/snowowl-{edition}_{version}/resources/*.sql` files after unpacking the installation archive. To load terminology data, save and execute the following script:

```
#!/usr/bin/env bash

#
# Copyright 2013-2017 B2i Healthcare Pte Ltd, http://b2i.sg
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

# Loads database content for Snow Owl from SQL dumps. Uses the name of the SQL
# file(s) specified as the target database for each file.
#
# Usage: ./snowowl_load_db [dbfile1] [dbfile2] ...

if [[ $# -eq 0 ]]; then
    echo "No SQL files were specified for loading. Exiting."
    exit 1
fi

MYSQL=$(which mysql)
BASENAME=$(which basename)
USER="snowowl"
PASSWORD="snowowl_pwd"

for DBFILE in "$@"
do
    DB="$(${BASENAME} "$DBFILE")"
    ${MYSQL} --batch -u${USER} -p${PASSWORD} "${DB%.sql}" < "${DBFILE}"

    if [ $? -ne 0 ]; then
        echo "Loading from file ${DB} failed."
    else
        echo "Loading from file ${DB} finished successfully."
    fi
done

echo "All files processed."
exit 0
```

Java

Download the “Linux X64” edition, and install it with yum:

```
# yum install jdk-8u121-linux-x64.rpm
```

LDAP

Install the latest (2.4.x) OpenLDAP with yum:

```
# yum -y install openldap compat-openldap openldap-clients openldap-servers openldap-servers-sql openldap-devel
```

Start the service, and enable it to run when system boots up:

```
# systemctl start slapd.service
# systemctl enable slapd.service
```

Next, run the `slappasswd` to create an LDAP root password. Please take note of this root password, the entire hashed value that is returned as output and starts {SSHA}, as you’ll use it throughout this article. Afterwards, you can import the provided LDIF files via `ldapadd/ldapmodify` commands (Role and Permission schema).

Configuring LDAP

Using LDIF dumps

B2i provided LDAP packages include the following content:

permission_schema.ldif

LDAP schema to use for authorization (contains definitions for permissions)

permissions.ldif

All available permissions in the system

roles.ldif

All available roles in the system

pm.ldif

Maps permissions to roles

update.sh

An update script using `ldapmodify` and `ldapadd` commands against a running LDAP instance to update it based on the files above

Optionally the assembly can contain two additional files:

users.ldif

All users available in the system

rm.ldif

Maps roles to users in the system

The update script will also make use of these files if any of them exist.

Install the `openldap-clients` first to make use of the script:

```
# yum install openldap-clients
```

Before updating the LDAP server, it is advised to shut down the service, and create a backup, so it can be restored easily if the script fails.

Restart the server, then create a new `ldif-<version>` folder and unzip the contents of the LDIF archive into this folder. Finally, execute the script to update the contents of LDAP:

```
# chmod u+x update.sh

# ./update.sh
Not specified LDAP URI parameter, using ldap://localhost:10389
adding new entry "cn=permission, ou=schema"
adding new entry "ou=attributeTypes, cn=permission, ou=schema"
...
modifying entry...
```

In case an error occurs, the executed command and the error response will be displayed. Errors will also be logged to a `{file_name}.errors` file, where the `{file_name}` refers to the file being processed (eg. `permissions.errors`).

When executing the script it is possible to get the following errors:

- `ERR_250_ALREADY_EXISTS` (or any synonym of `ALREADY_EXISTS`)
- `ERR_54 Cannot add a value which is already present : snomed:compare:automap`
- `ERR_335 Oid 2.25.128424792425578037463837247958458780603.1 for new schema entity is not unique`

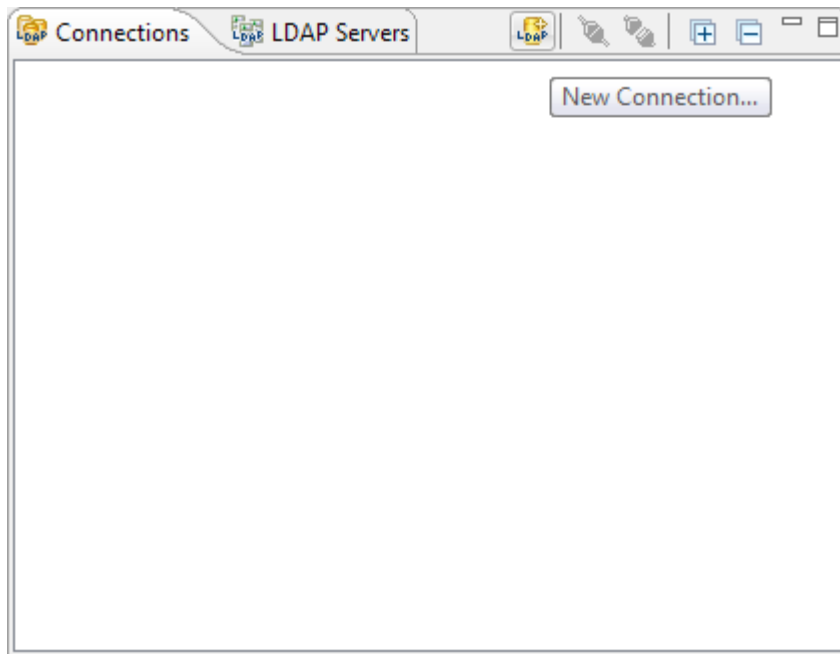
This is expected as most of the time the LDAP instance will already contain an existing definition of some entries and/or schema entities. If you notice other errors (either during script execution or when using the LDAP), roll back your instance to a previous state from a backup.

By default the update script will execute against the LDAP instance running locally at `ldap://localhost:10389`; if you'd like to run the script against a remote LDAP server (or the LDAP is listening on a different port), you can do it by specifying the `LDAP_URI` parameter:

```
# ./update.sh ldap://<host>:<port>
```

Using Apache Directory Studio

Open Apache Directory Studio, create a new connection using the first button on the “Connections” toolbar:



Enter connection name, hostname, and port, then hit **Next >** to go to the next page in the wizard:

New LDAP Connection

Network Parameter

Please enter connection name and network parameters.

Connection name: snowowl

Network Parameter

Hostname: localhost

Port: 10389

Encryption method: No encryption

Server certificates for LDAP connections can be managed in the '[Certificate Validation](#)' preference page.

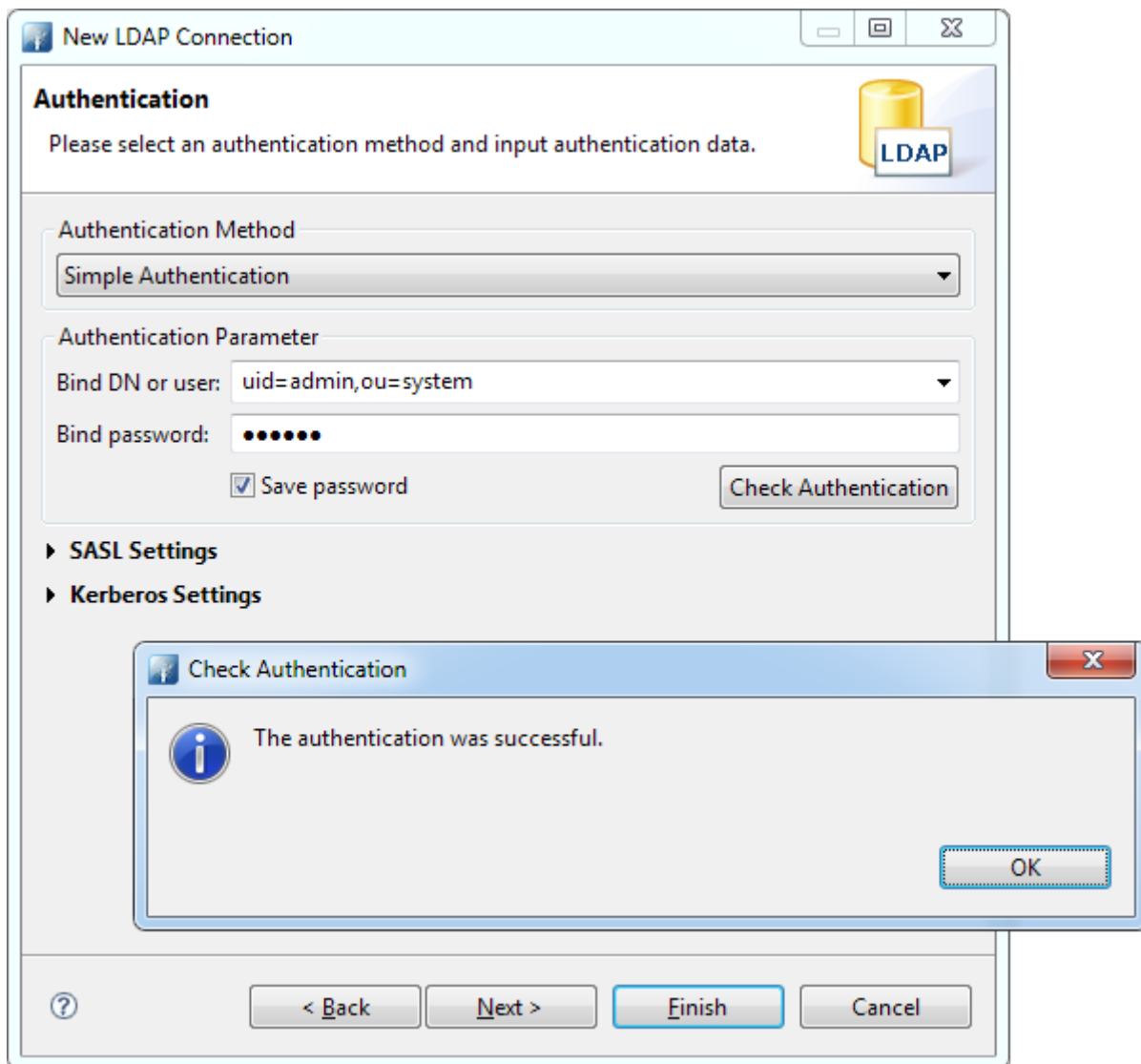
Provider: Apache Directory LDAP Client API

Check Network Parameter

☐ Read-Only (prevents any add, delete, modify or rename operation)

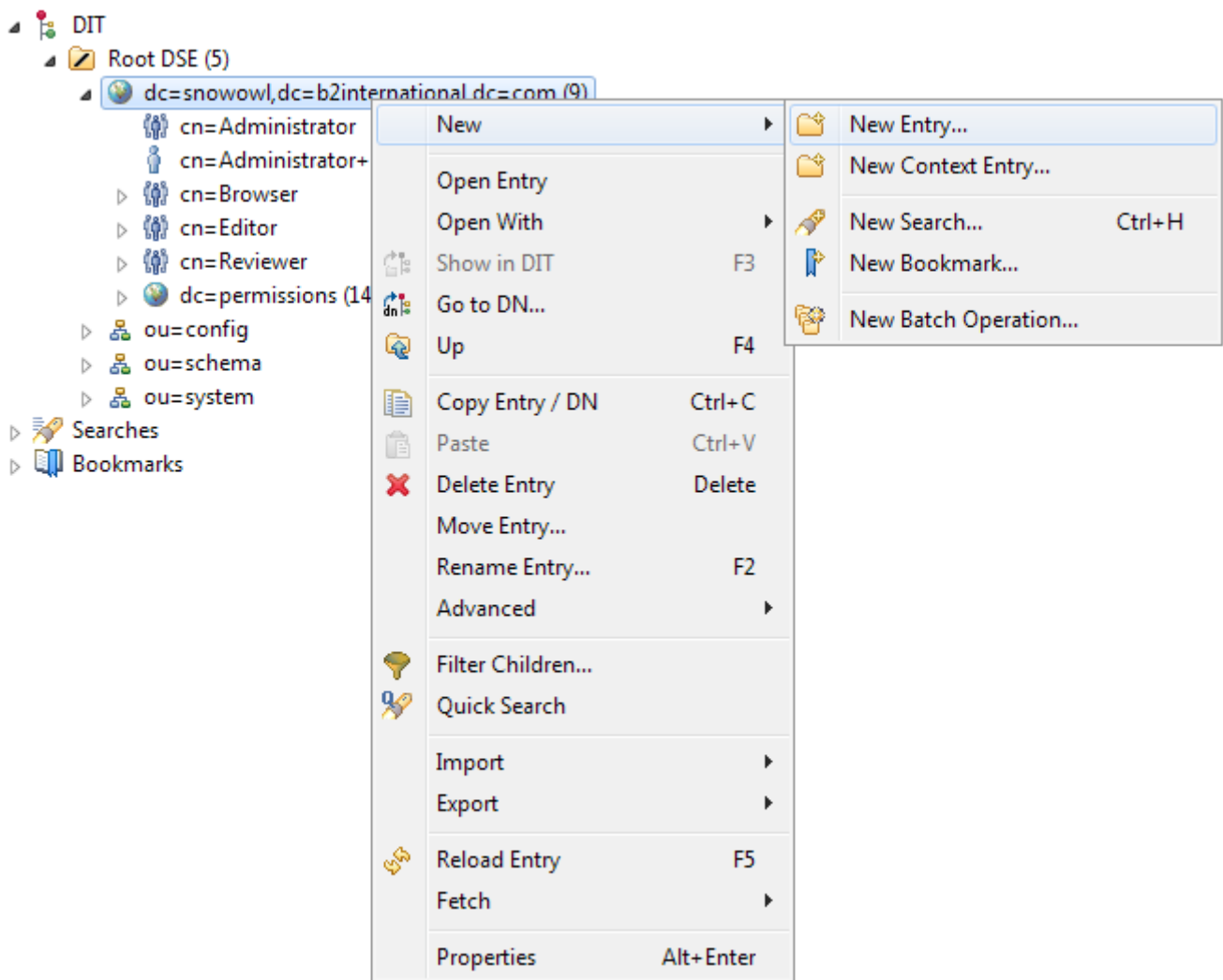
? < Back Next > Finish Cancel

Set authentication method to **Simple Authentication**, enter Bind DN of your root user and its password. Click **Check Authentication** to make sure the values are accepted, then dismiss the wizard with **Finish**:

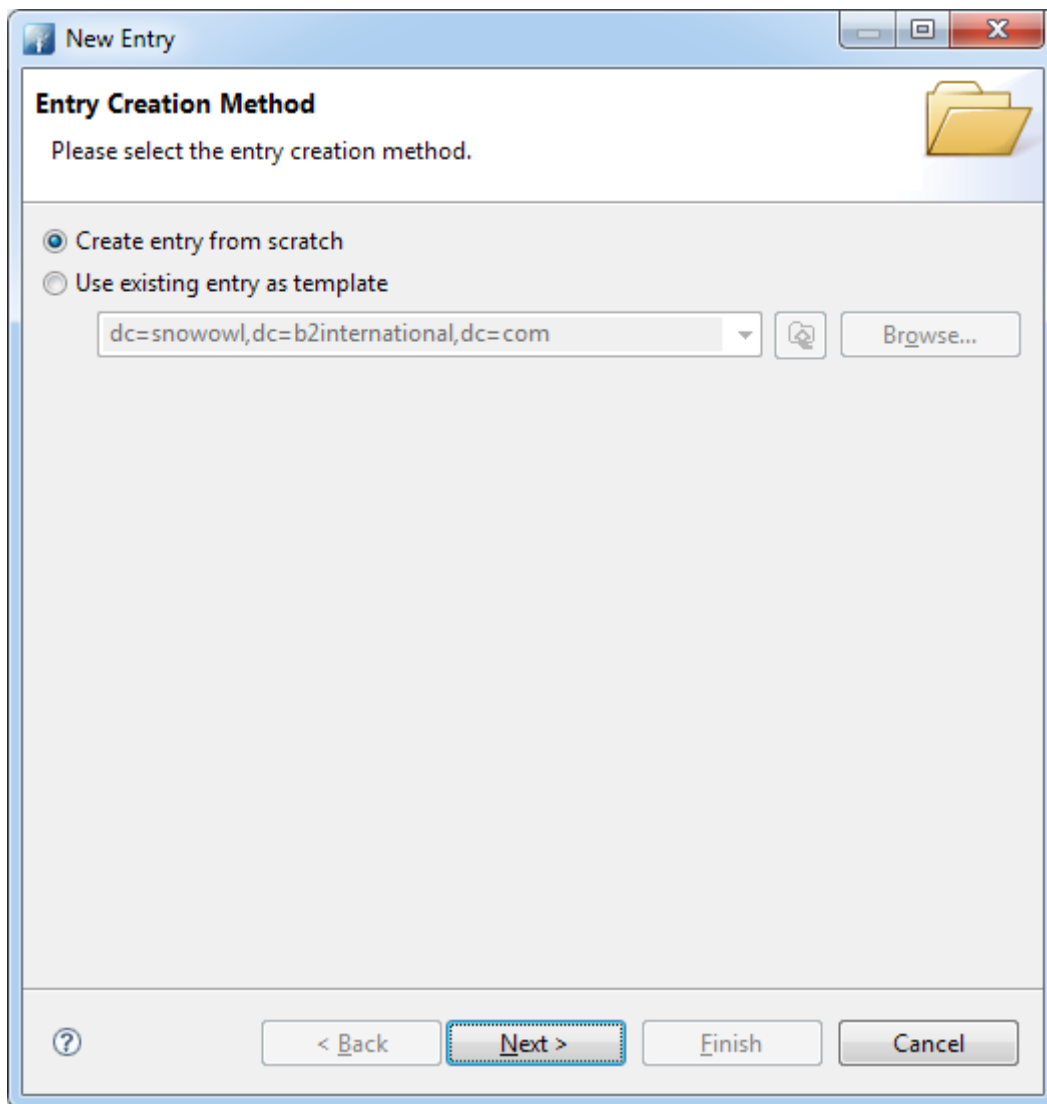


Creating a new user using Apache Directory Studio

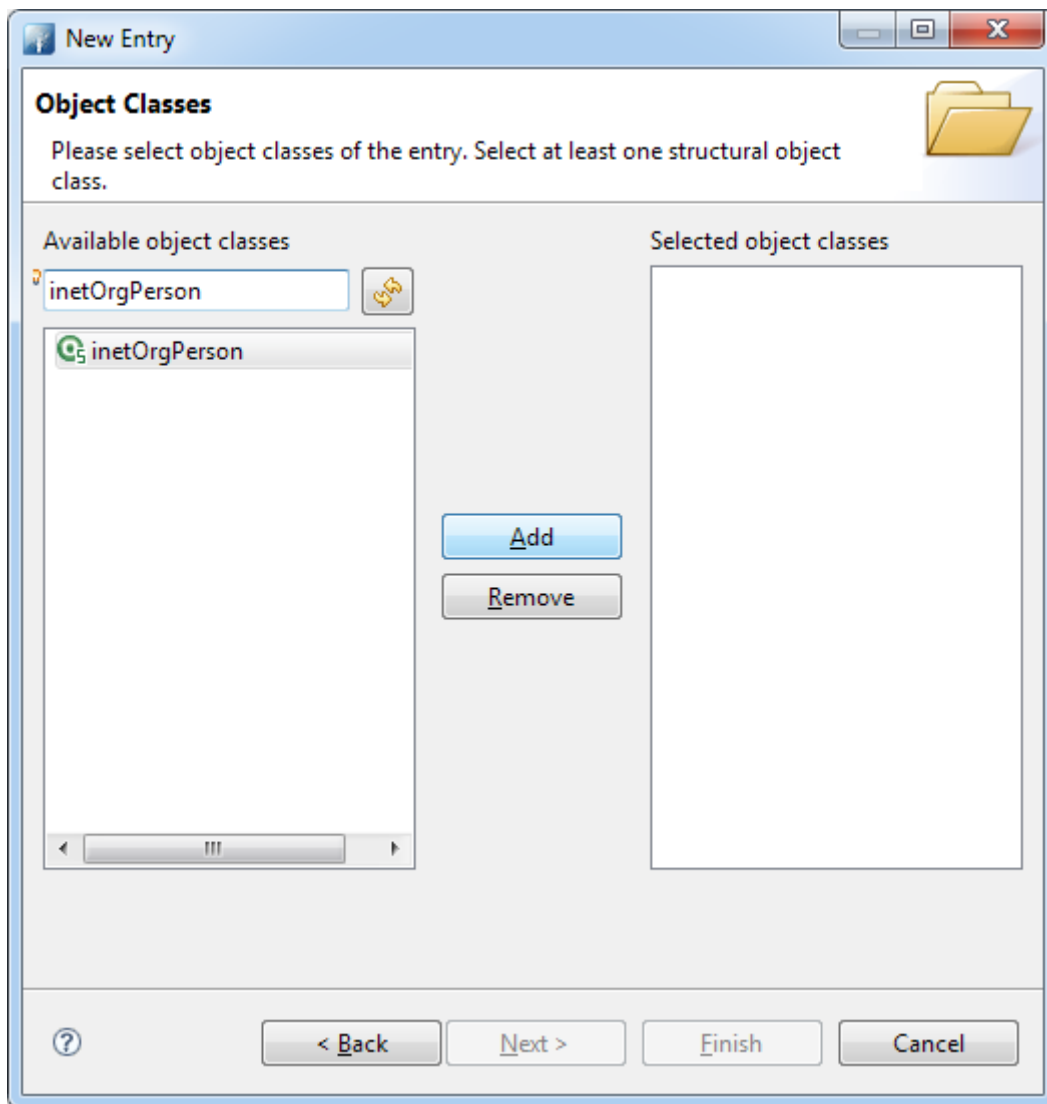
Go to LDAP Browser view, right click on the Domain component (DC) and add new entry via **New > New entry**:



Create a new entry from scratch:



Select **inetOrgPerson** object from the wizard, add it as a selected object class:




Configure the Relative Distinguished Name (RDN). Specify the common name (CN), surname (SN) and unique ID (uid):

New Entry

Distinguished Name


Please select the parent of the new entry and enter the RDN.

Parent: 

RDN:

<input type="text" value="cn"/>	=	<input type="text" value="Snow Owl User"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="sn"/>	=	<input type="text" value="info"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="uid"/>	=	<input type="text" value="info@b2international.com"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

DN Preview:



Open the added node in an editor, right click in the editor and select **New Attribute**:

cn=Snow Owl User+sn=info+uid=info@b2international.com,dc=snowowl,dc=b2international,dc=com

DN: cn=Snow Owl User+sn=info+uid=info@b2international.com,dc=snowowl,dc=b2international,dc=com

Attribute Description	Value
<i>objectClass</i>	<i>inetOrgPerson (structural)</i>
<i>objectClass</i>	<i>organizationalPerson (structural)</i>
<i>objectClass</i>	<i>person (structural)</i>
<i>objectClass</i>	<i>top (abstract)</i>
cn	Snow Owl User
sn	info
uid	info@b2international.com

- New Attribute... Ctrl+Shift++
- New Value Ctrl++
- New Search... Ctrl+H
- New Batch Operation...
- Locate DN in DIT F3
- Open Schema Browser
- Show In
- Copy Value Ctrl+C
- Paste Ctrl+V
- Delete Value Delete
- Select All Ctrl+A
- Advanced
- Edit Attribute Description F6
- Edit Value F7
- Edit Value With
- Edit Entry... F8
- Reload Attributes F5
- Fetch Operational Attributes
- Properties Alt+Enter

Select **userPassword** attribute, click **Finish**, then enter user password in the **Password Editor** dialog:

Password Editor

New Password

Enter New Password: new_user_pwd

Select Hash Method: SSHA-512

Password Preview: {SSHA-512}vtN64f6/EPhtfsx9ozxR9zko0GoV8ZQb6ND2sWSu5qHmi4e0l4rYtHgO9noSyXaldj1IdE

New Salt

Password (Hex): bed37ae1febf10f86bb5fb31f68cf147dce4a741a857c6506fa343dac

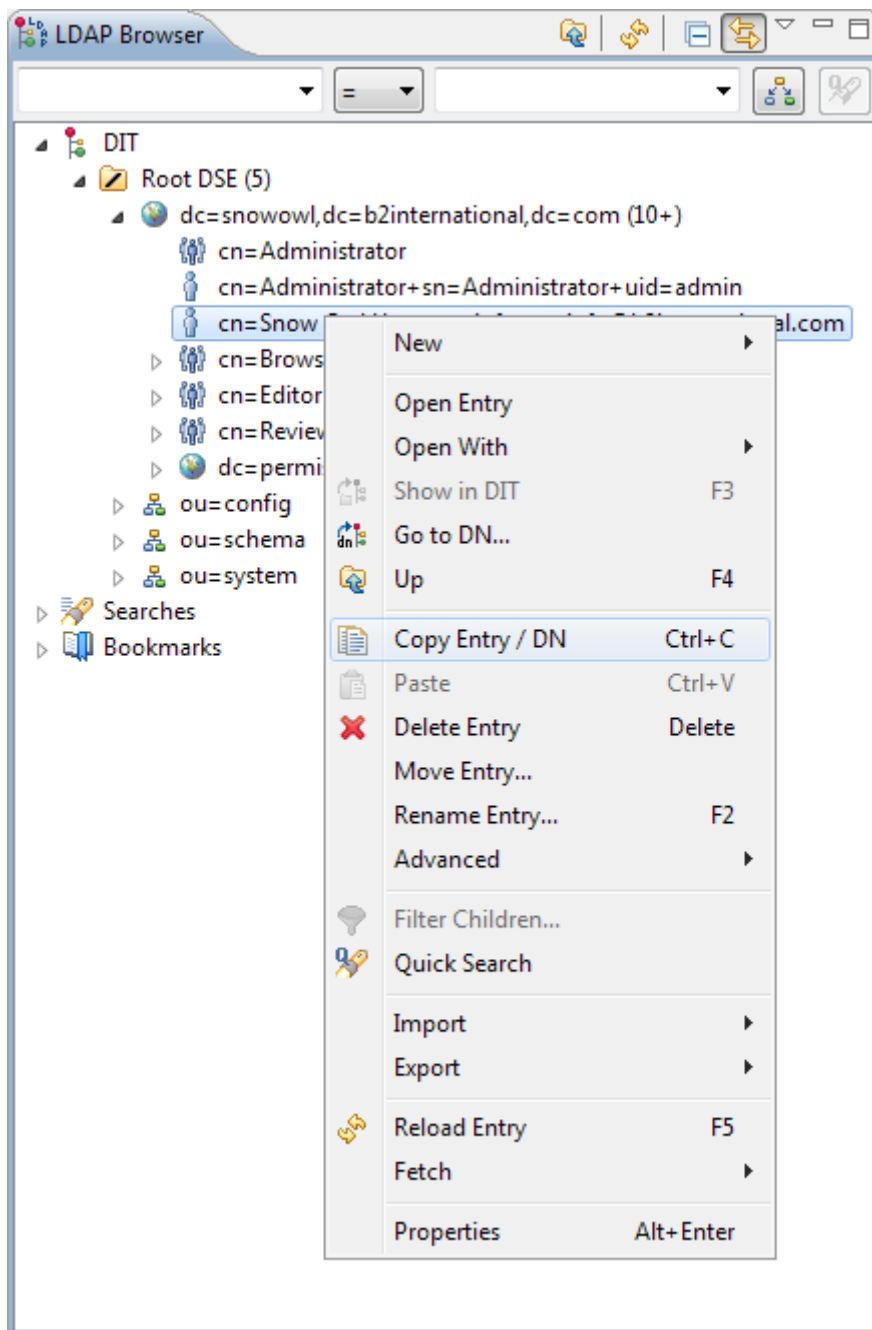
Salt (Hex): 5266ebc683682a35

☒ Show new password details

OK Cancel

Finally, add a `uniqueMember` attribute to the Administrator group.

Select the new user's node in the tree, right click and select `Copy Entry / DN:`



Right click the `uniqueMember` attribute of the `Administrator` node, select `New Value` and paste the previously copied DN of the new user as the value:

DN: cn=Administrator,dc=snowowl,dc=b2international,dc=com

Attribute Description	Value
<i>objectClass</i>	<i>groupOfUniqueNames (structural)</i>
<i>objectClass</i>	<i>role (structural)</i>
<i>objectClass</i>	<i>top (abstract)</i>
cn	Administrator
▶ permissionId (141 values)	
uniqueMember	cn=Administrator+sn=Administrator+uid=admin,dc=snowowl,dc=b2international,dc=com
uniqueMember	cn=Snow Owl User+sn=info+uid=info@b2international.com,dc=snowowl,dc=b2international,dc=com

The operation takes place after pressing Enter or removing the focus from the edited field.

Snow Owl Server

Unpack an official distribution archive into `/opt`, installing `unzip` first if not already present; change permissions on the created folder:

```
# yum install unzip
# unzip snowowl-{edition}-{version}-mysql.zip -d /opt
# chown -Rv snowowl:snowowl /opt/snowowl-{edition}-{version}
```

Update Snow Owl's Configuration

Update `snowowl_config.yml` to use LDAP as identity provider and set the MySQL password for `snowowl`, created earlier:

```
identity:
  providers:
    - ldap:
        uri: ldap://<host>:<port>
        baseDn: <your-base-dn>
        rootDn: <DN of the ROOT user>
        rootDnPassword: <password of the ROOT user>
        usePool: false

repository:
  ...

  database:
    ...
    username: snowowl
    password: snowowl ①
```

① Update MySQL username and password, if necessary

Memory settings

Heap size used by Snow Owl can be adjusted in `dmk.sh`; look for the following section:

```
JAVA_OPTS="$JAVA_OPTS \
-Xms12g \
-Xmx12g \
```

`Xms` sets the minimum heap size, `Xmx` sets the maximum heap size used by the JVM.

OSGi console

The OSGi console can be accessed both via `ssh` and `telnet`. Configuration settings for remote access can be found in `osgi.console.properties`. The default settings are:

`/opt/snowowl-{edition}_{version}/repository/ext/osgi.console.properties`

```
telnet.enabled=true
telnet.port=2501
telnet.host=localhost
ssh.enabled=true
ssh.port=2502
ssh.host=localhost
```

Further information on how to enable/disable the OSGi console can be found here: <http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.4.RELEASE/docs/virgo-user-guide/html/ch08.html>.

For opening a telnet connection to the server, type:


```
$ telnet localhost 2501
Trying ::1...
Connected to localhost.
Escape character is '^]'.
osgi>
```

Logging

Log files are stored under `./opt/snowowl-{edition}_{version}/serviceability` directory of the Snow Owl server. The following log files are created:

`logs/log.log`

Generic system trace log file, all log messages are written into this file. Logs files are created for each day with the following file name format `log_%d{yyyy-MM-dd}`. Snow Owl will keep 90 days worth of history in this folder before starting to remove files. This log serves two main purposes:

1. It provides global trace files that capture high-volume information regarding the Virgo's internal events. The files are intended for use by support personnel to diagnose runtime problems.
2. It provides application trace files that contain application-generated output. This includes output generated using popular logging and tracing APIs including the OSGi LogService, as well as output generated by calls to `System.out` and `System.err`. These files are intended for use by application developers and system administrators. An application is defined as a scope so a single bundle will not get its own log file unless it is a Web application Bundle or is included in a scoped plan or a par file.

`logs/access/*.log`

Web container access log files in the same format as those created by standard web servers. The log files are prefixed with the string `localhost_access_log`, have a suffix of `.txt`, use a standard format for identifying what should be logged, and do not include DNS lookups of the IP address of the remote host.

`eventlogs/eventlog.log`

The `EVENT_LOG_FILE` appender logs only important events and thus the volume of information is lower.

`logs/snowowl/snowowl_user_audit.log`

Events with business significance will be logged in this file.

`logs/snowowl/snowowl_user_access.log`

User access events are logged in this log file. Both authorized and unauthorized access is logged.

`logs/snowowl/snowowl_import.log`

Import processes log into this file detailed information about import.

`logs/snowowl/snowowl_export.log`

Export processes log into this file detailed information about export.

Detailed information on the configuration on the logging configuration can be found [here](#):

<http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.4.RELEASE/docs/virgo-user-guide/html/ch11.html>.

Currently, default logging appenders for the log targets above look like this:

```
<appender name="LOG_FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>serviceability/logs/log.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>serviceability/logs/log_%d{yyyy-MM-dd}.log</fileNamePattern>
    <!-- keep 90 days' worth of history -->
    <maxHistory>90</maxHistory>
  </rollingPolicy>
  <filter class="ch.qos.logback.core.filter.EvaluatorFilter">
    <evaluator class="ch.qos.logback.classic.boolex.OnMarkerEvaluator">
      <marker>SNOW_OWL_USER_ACCESS</marker>
    </evaluator>
    <onMismatch>ACCEPT</onMismatch>
    <onMatch>DENY</onMatch>
  </filter>
  <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
    <Pattern>[%d{yyyy-MM-dd HH:mm:ss.SSS}] %-5level %-28.28thread %-
64.64logger{64} %X{medic.eventCode} %msg %ex%n</Pattern>
  </encoder>
</appender>
```

In this setting, the administrator can set the location of the log file, the maximum size of the log file and the total number of files rolling over. Documentation on the logging configuration settings can be found here: <http://logback.qos.ch>.

Web Server Configuration

Snow Owl Server uses Tomcat as its built-in web server for administrative and RESTful services. The configuration settings for the web server can be found in `tomcat-server.xml`. Detailed information on configuring the different elements can be found here: <http://tomcat.apache.org/tomcat-7.0-doc/config/index.html>. The most important settings are the port numbers for HTTP and HTTPS protocols:

/opt/snowowl-{edition}_{version}/configuration/tomcat-server.xml

```
<Service name="Catalina">
  <Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="configuration/keystore"
    keystorePass="changeit"/>
```

Web Server Administrative Console application

The Admin Console is a web application for managing the Virgo Server instance powering Snow Owl Server. The default location of the admin console is at <http://localhost:8080/admin>.

The Admin Console is a password-protected page; to configure users allowed to access the Admin Console, change settings in file `org.eclipse.virgo.kernel.users.properties`. The username-password pair configured by default is `user=admin`, `pwd=adminpwd`:

/opt/snowowl-{edition}_{version}/configuration/org.eclipse.virgo.kernel.users.properties

```
#####
# User definitions
#####
user.admin=adminpwd

#####
# Role definitions
#####
role.admin=admin
```

More information on administrative user access control can be found on the following pages: <http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.4.RELEASE/docs/virgo-user-guide/html/ch09.html> and <http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.4.RELEASE/docs/virgo-user-guide/html/ch13s06.html#configuring-authentication>.

Virgo documentation

Complete documentation of the Virgo OSGi server can be found here: <http://www.eclipse.org/virgo/documentation>.