Configuration guide

Introduction

This document helps configuring each installed component for use with Snow Owl Server.

Components

Operating system

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

WWW (HTTP)
: For accessing Bugzilla's SOAP API and web-based UI

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

11389/TCP
: Used by the LDAP server

Database server software

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/5.6/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
```

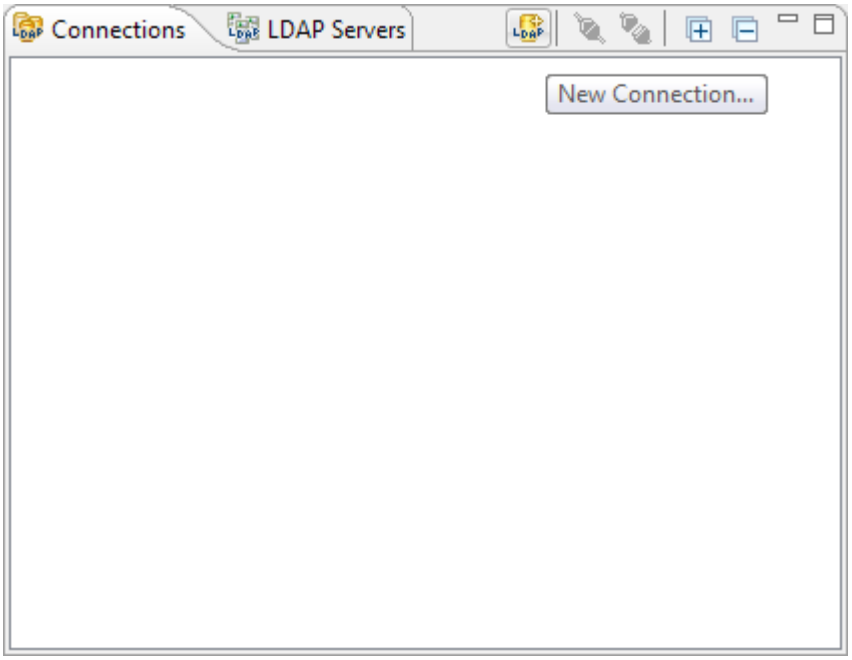
Restart the mysql service to make sure changes are picked up:

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

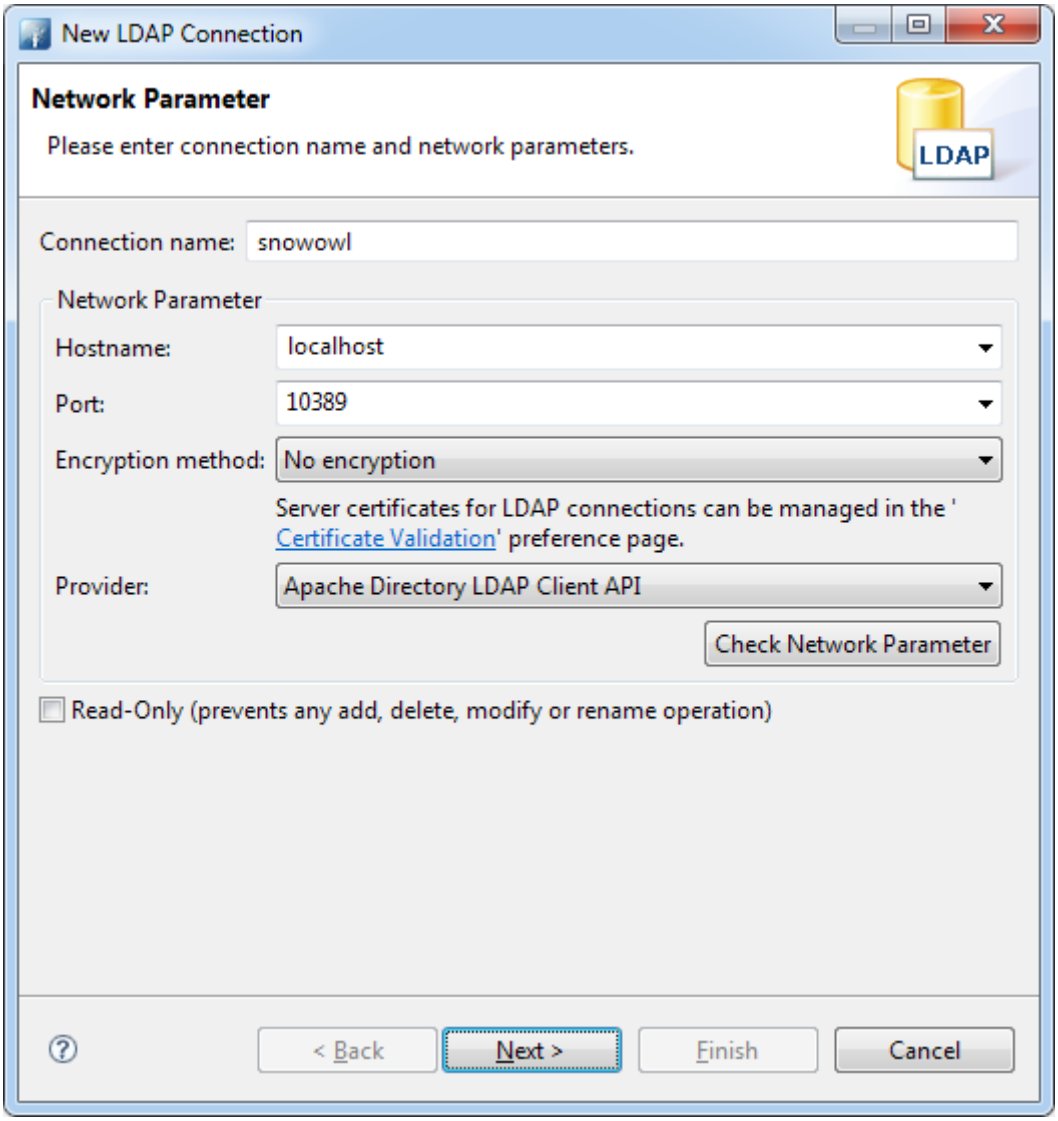
LDAP

Setting up a connection from Apache Directory Studio

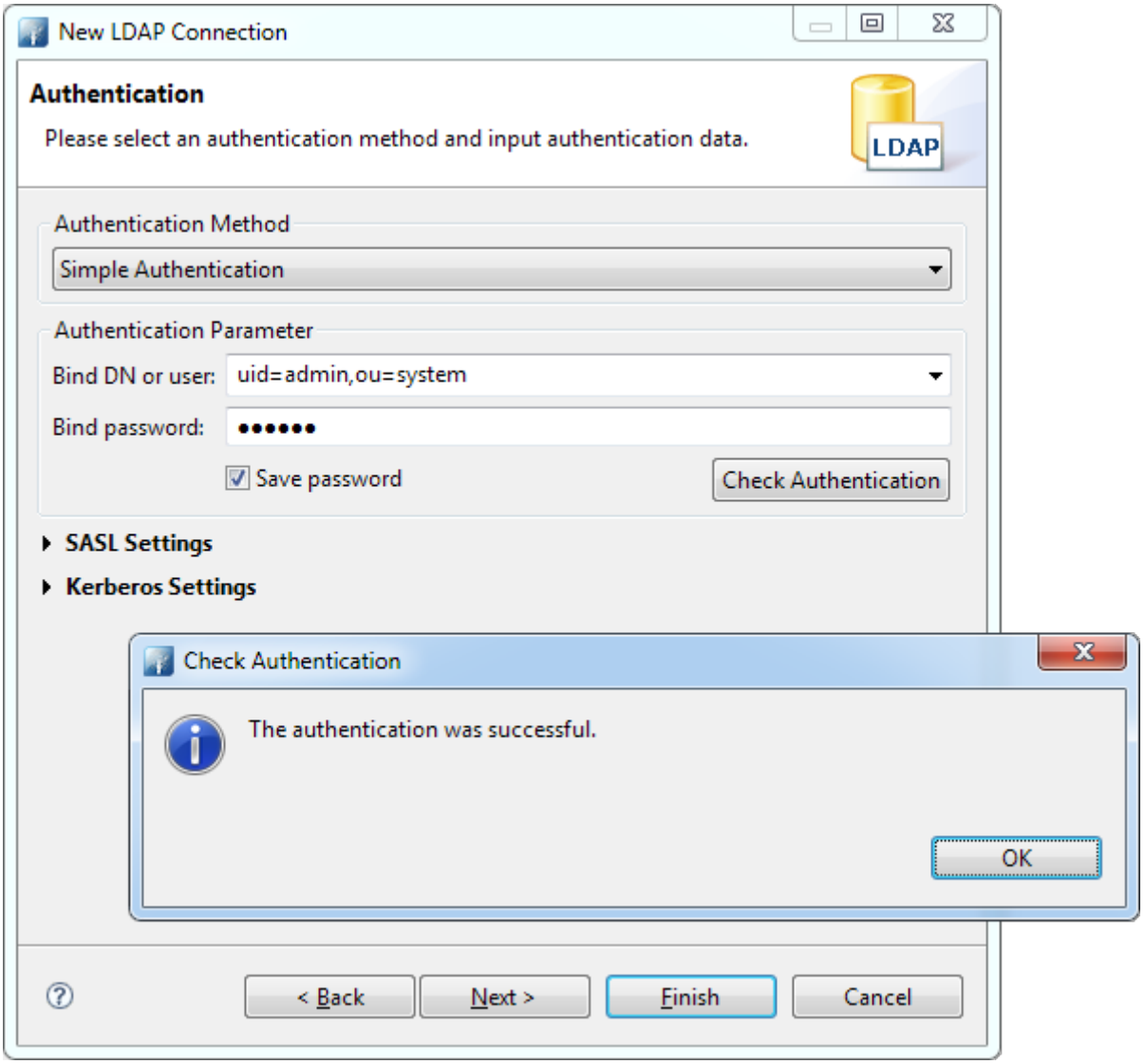
Open Apache Directory Studio, create a new connection using the first button on the “Connections” toolbar:



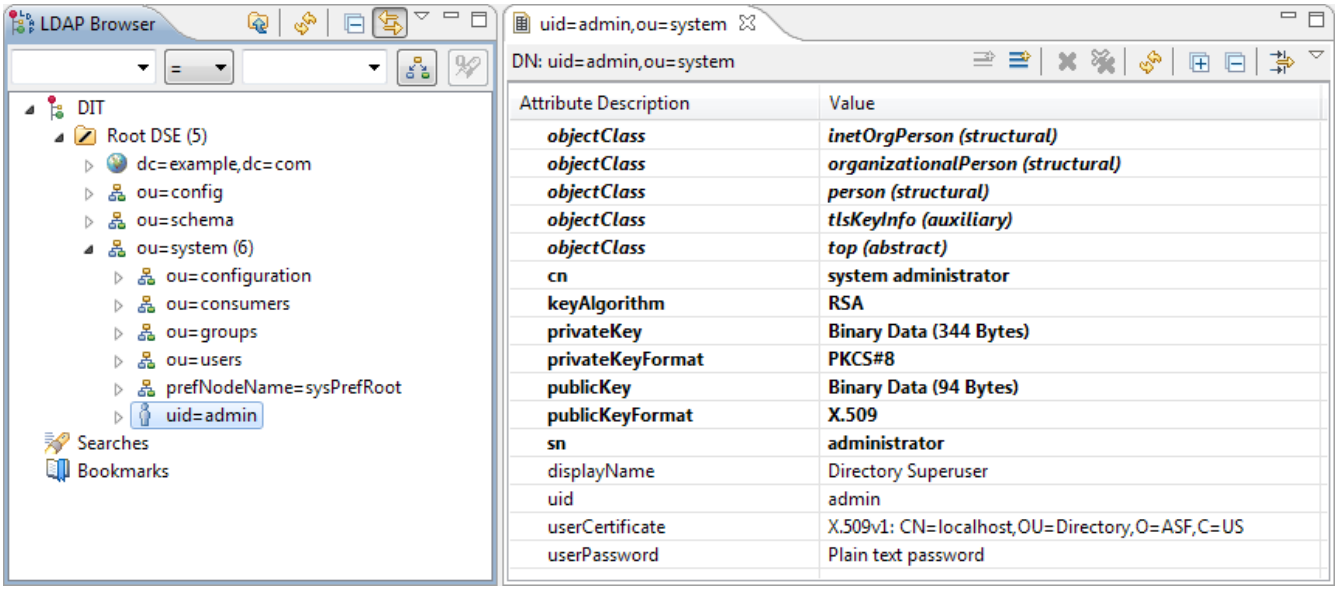
Enter connection name, hostname, and port, then hit `Next >` to go to the next page in the wizard:



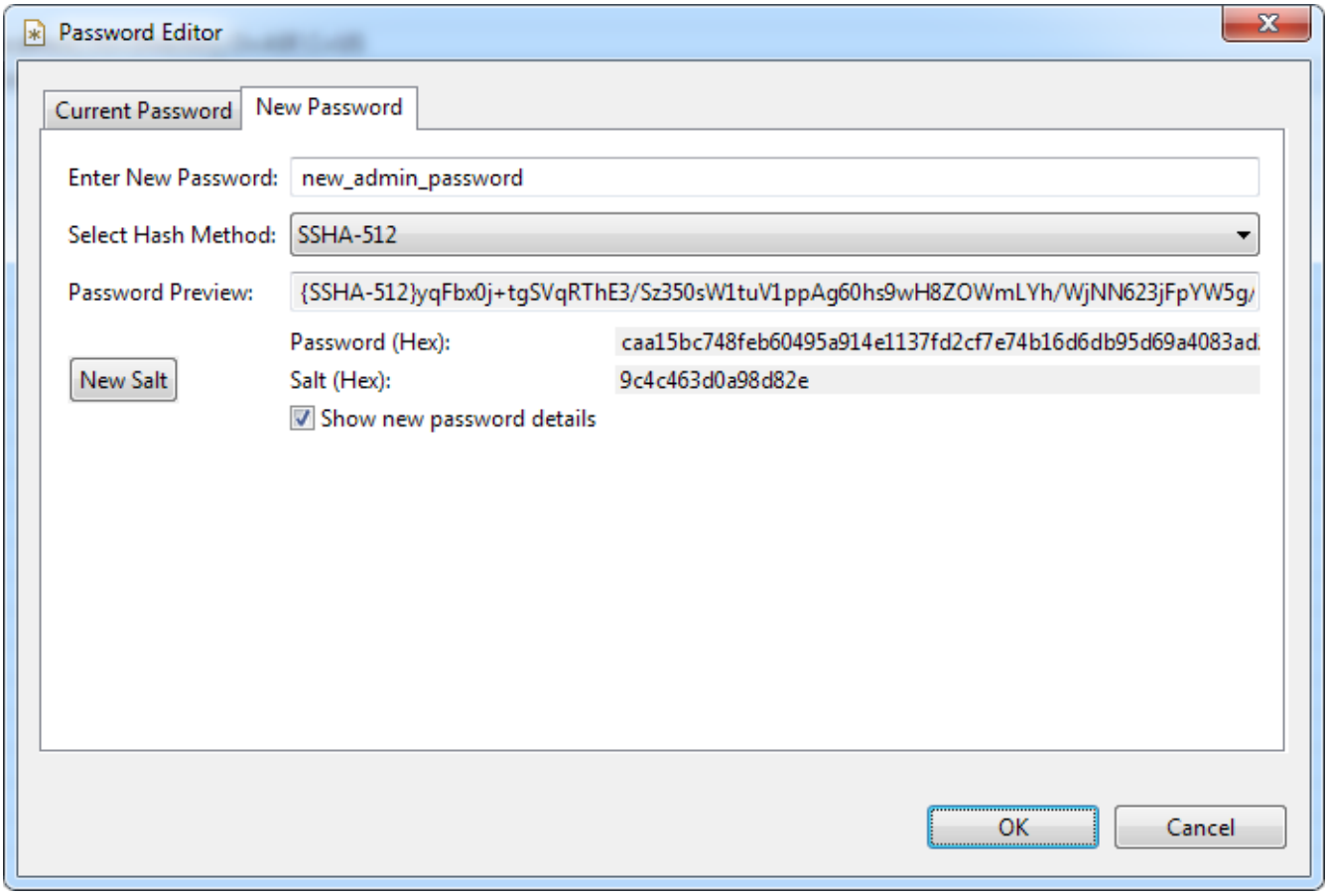
Set authentication method to `Simple Authentication`, enter Bind DN `uid=admin,ou=system` and password `secret`. Click `Check Authentication` to make sure the values are accepted, then dismiss the wizard with `Finish`:



After connecting, you are advised to change the default password for user `system`. In order to do so, expand node `ou=system` and select the person `admin`. The editor will now display related user information:



Double click on `userPassword` to open the password change dialog, select `SSHA-512` as the hash method. To see the new password as it is being entered, check `Show next password details`:

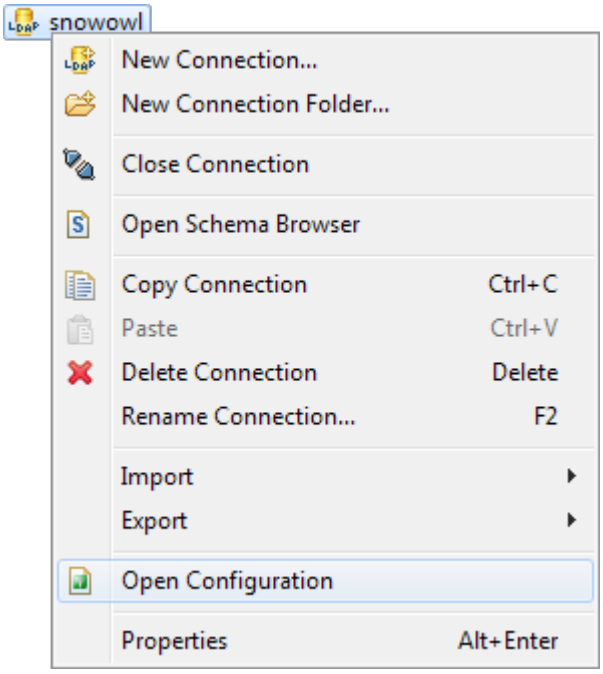


To see if the password was changed successfully:

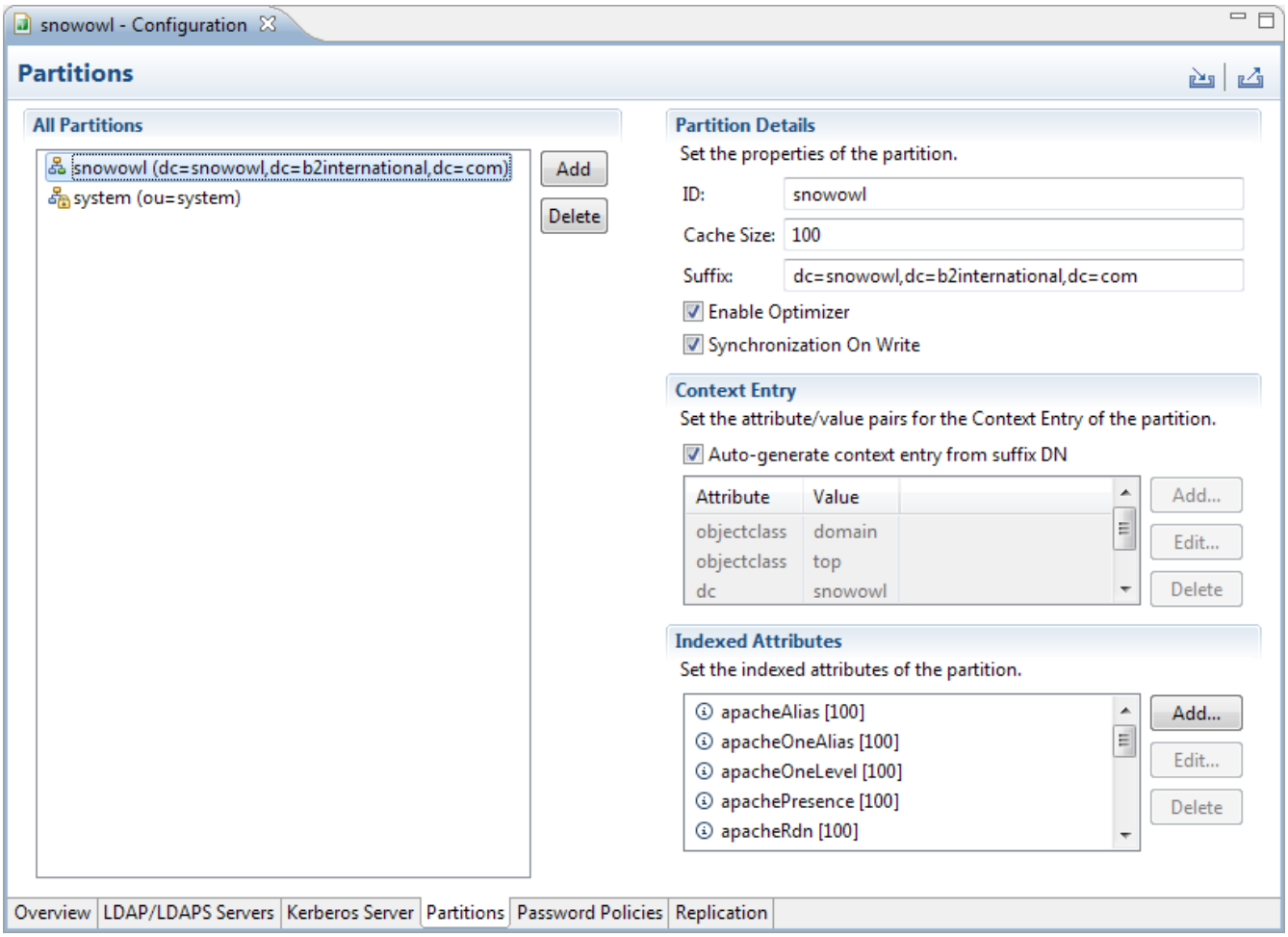
- Close the connection using the third button on the lower left toolbar named `Close Connection`,
- Right-click the item representing the connection to open its properties with the context menu item at the bottom,
- Change the previously entered default bind password to the updated value on the `Authentication` tab,
- Re-connect to the server using `Open Connection`.

Creating a partition for Snow Owl Server

Right-click on the item representing the connection, and select "Open Configuration":



An editor opens with the server configuration; select the `Partitions` tab. An `examples` partition is added by default, which should be removed. Add a new partition named `snowowl` and fill out the details (ID and suffix):



Save the editor and restart the server to apply changes in partitions.

```
# service apacheds-2.0.0_M12-default restart
Stopping ApacheDS - default...
Stopped ApacheDS - default.
Starting ApacheDS - default...
```

Using LDIF dumps

B2i provided LDAP packages include the following content:

schema.ldif
: LDAP schema to use for authorization (contains definitions for permissionId and role)

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 from the contents of folder `/var/lib/apacheds-2.0.0_M12/default`, 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`

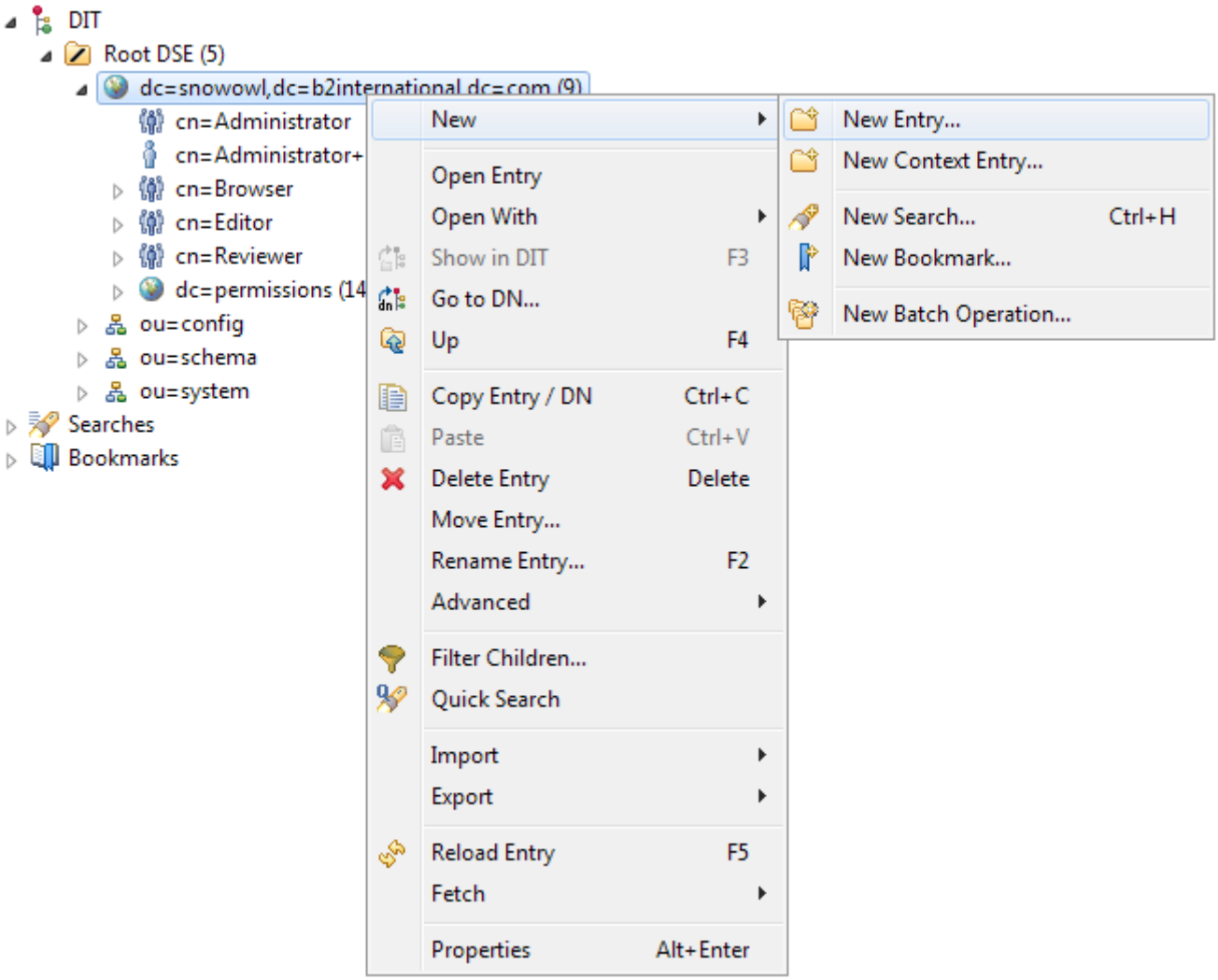
This is expected as most of the time the LDAP instance will already contain an existing definition of some entries and/or schema entities. If you notice other errors (either during script execution or when using the LDAP), roll back your instance to a previous state from a backup.

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:

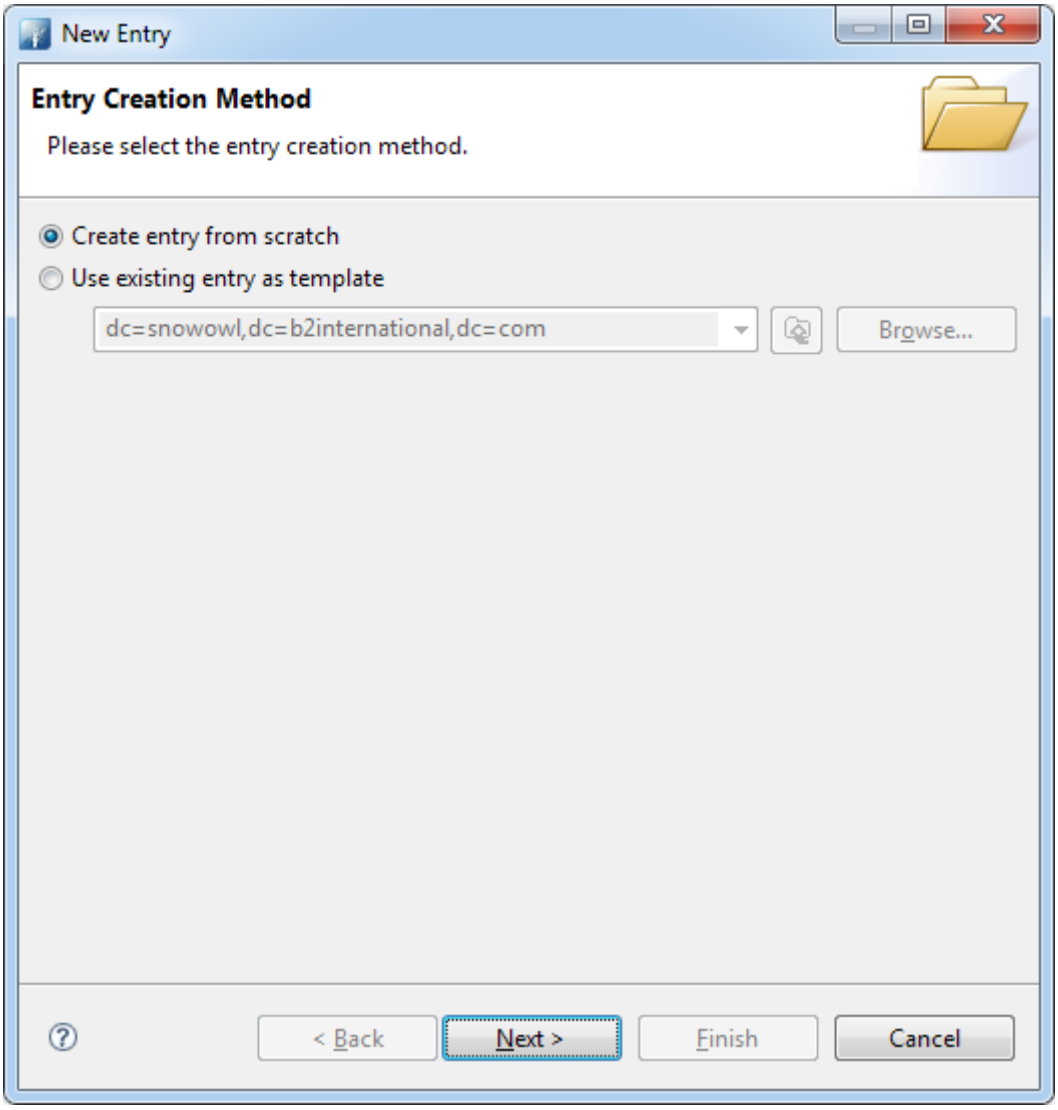
```
# ./update.sh ldap://<host>:<port>
```

Creating a new user from Directory Studio

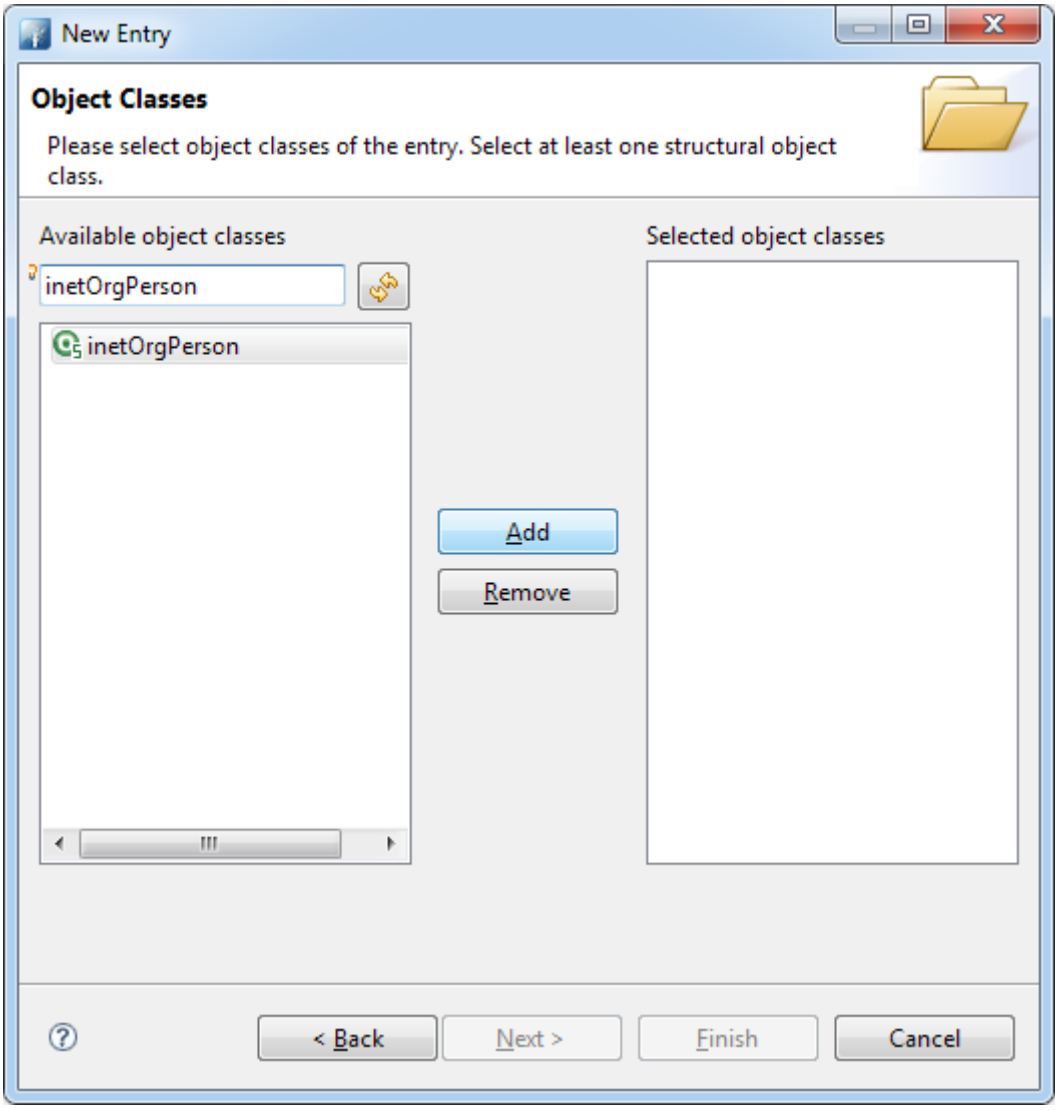
Go to LDAP Browser view, right click on the Domain component (DC) and add new entry via `New` > `New entry`:



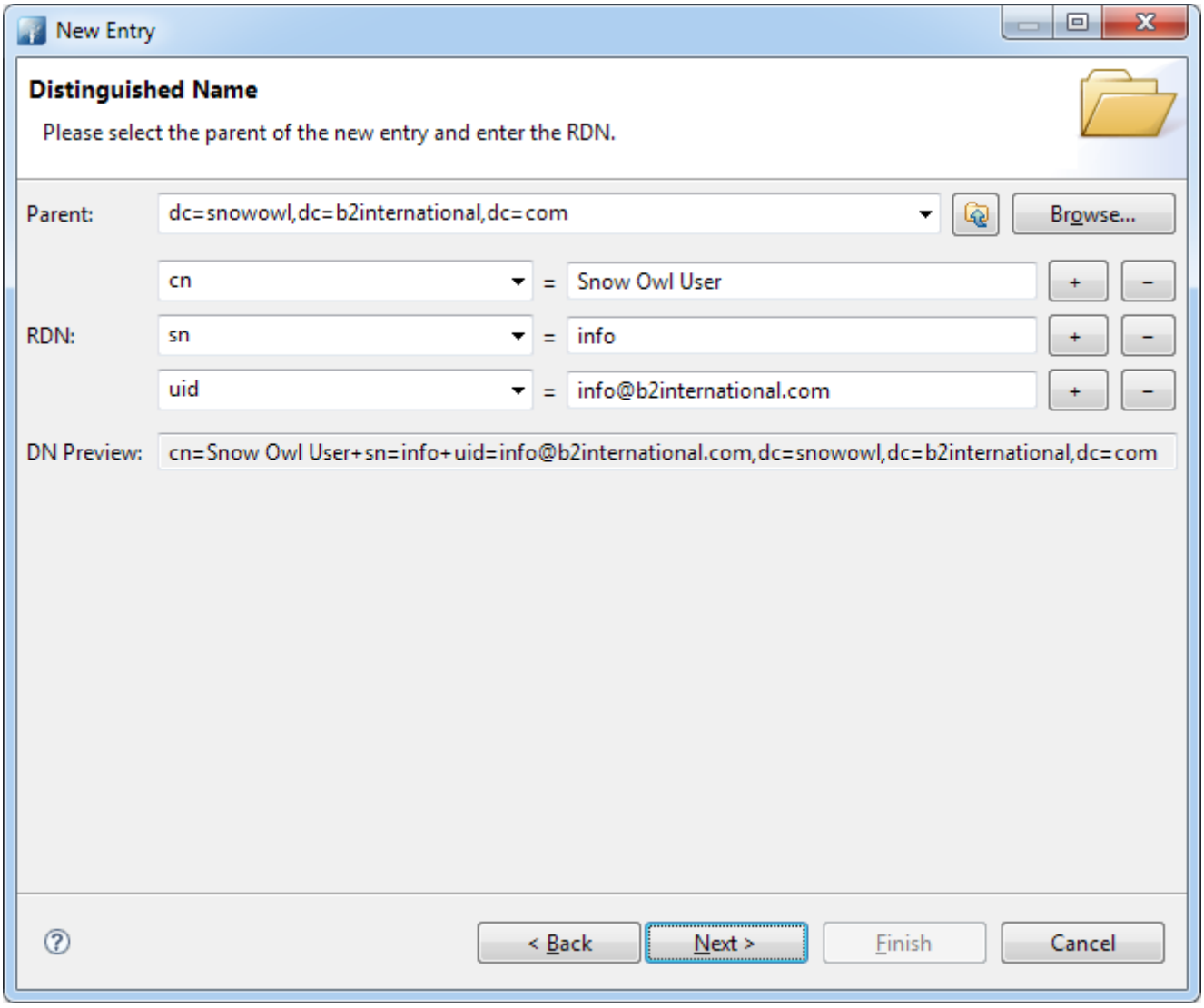
Create a new entry from scratch:



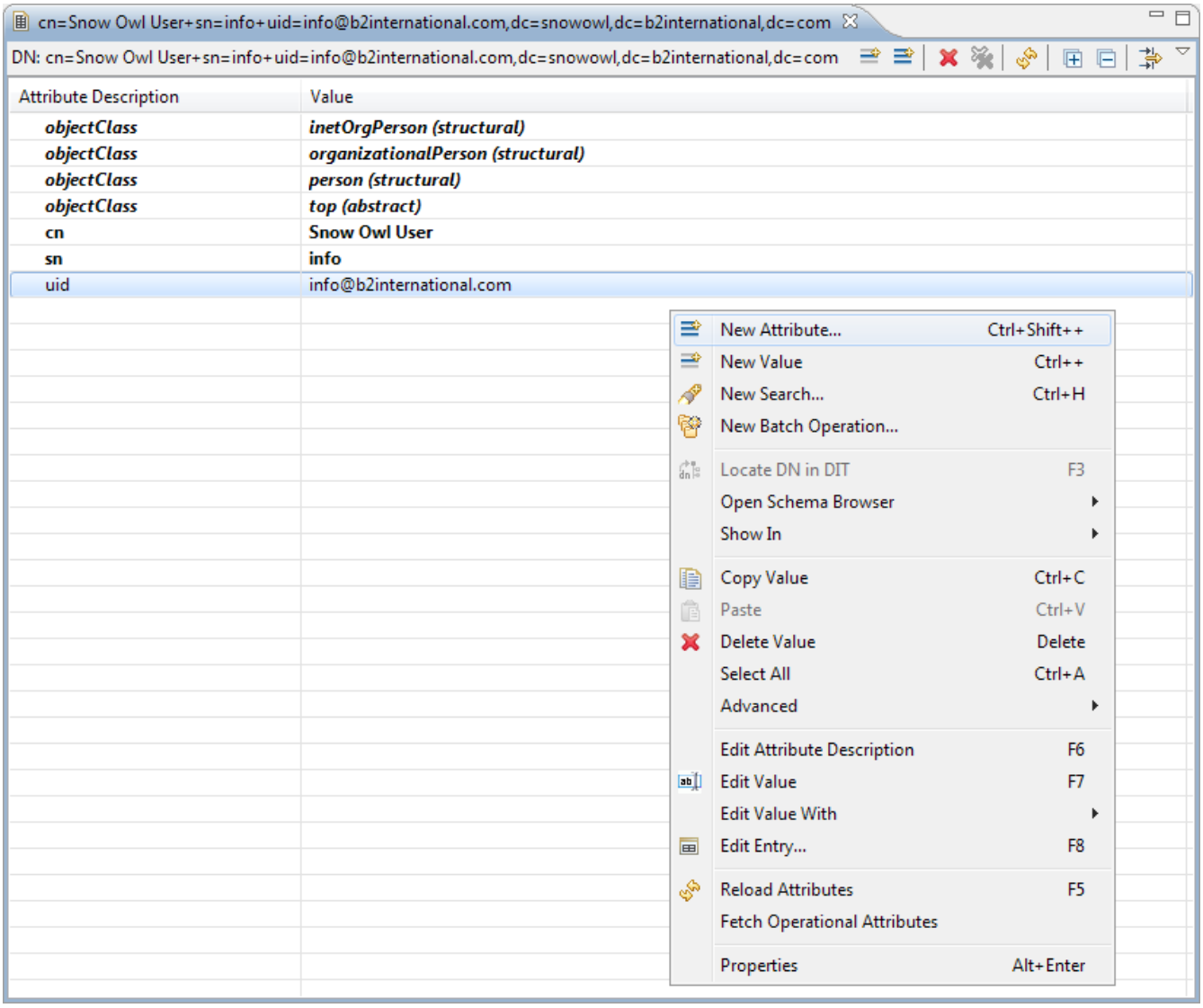
Select `inetOrgPreson` object from the wizard, add it as a selected object class:



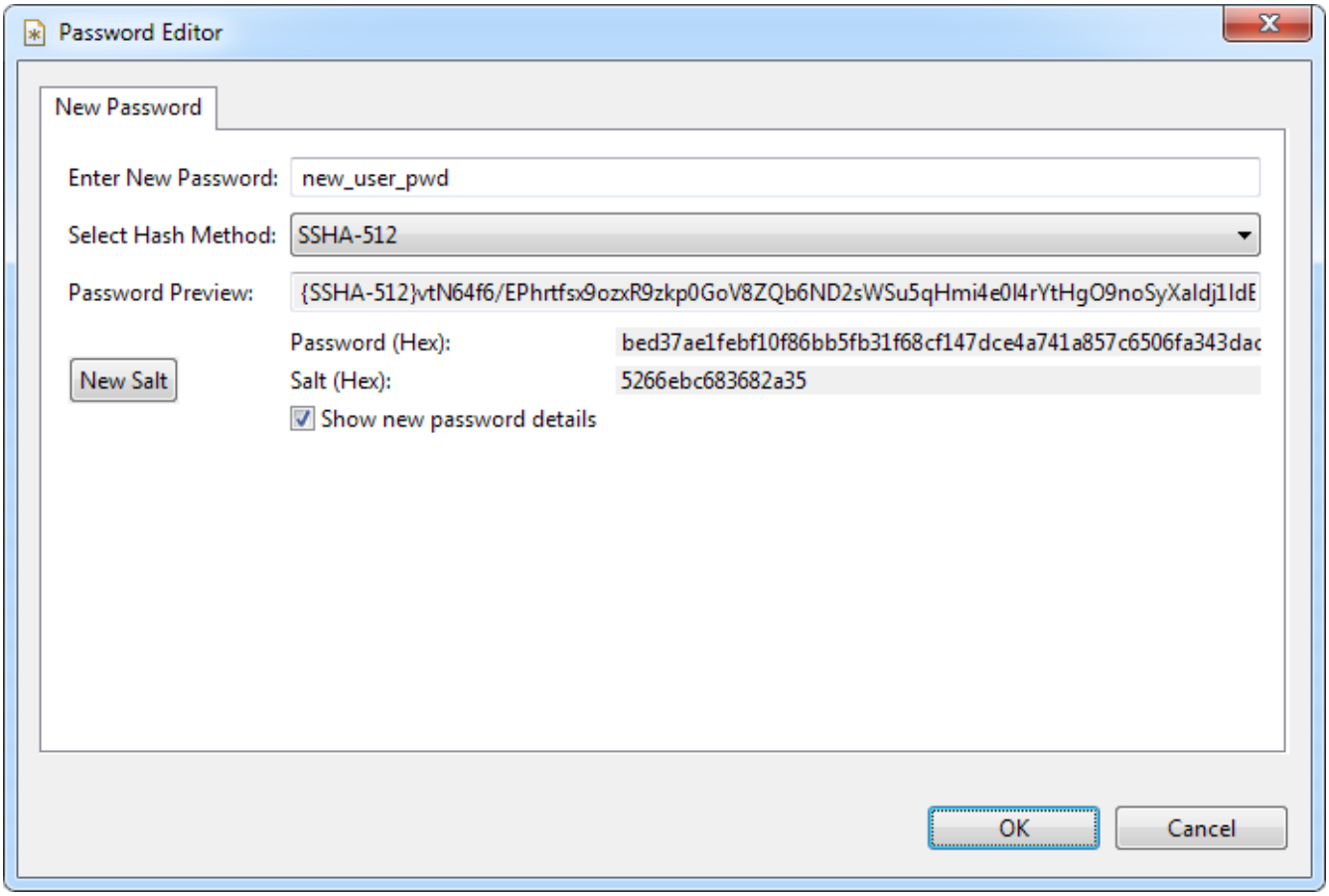
Configure the Relative Distinguished Name (RDN). Specify the common name (CN), surname (SN) and unique ID (uid):



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

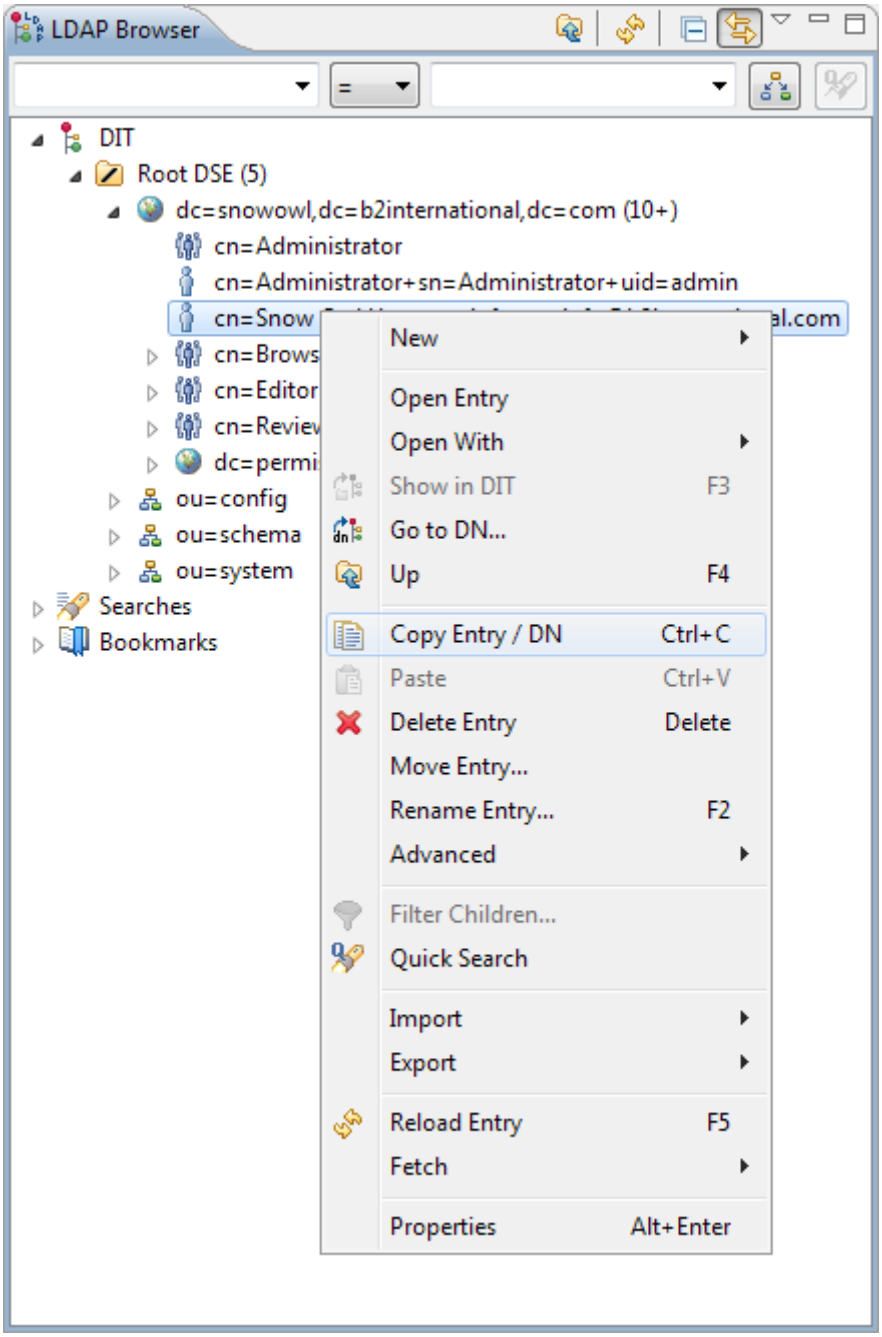


Select `userPassword` attribute, click `Finish`, then enter user password in the `Password Editor` dialog:

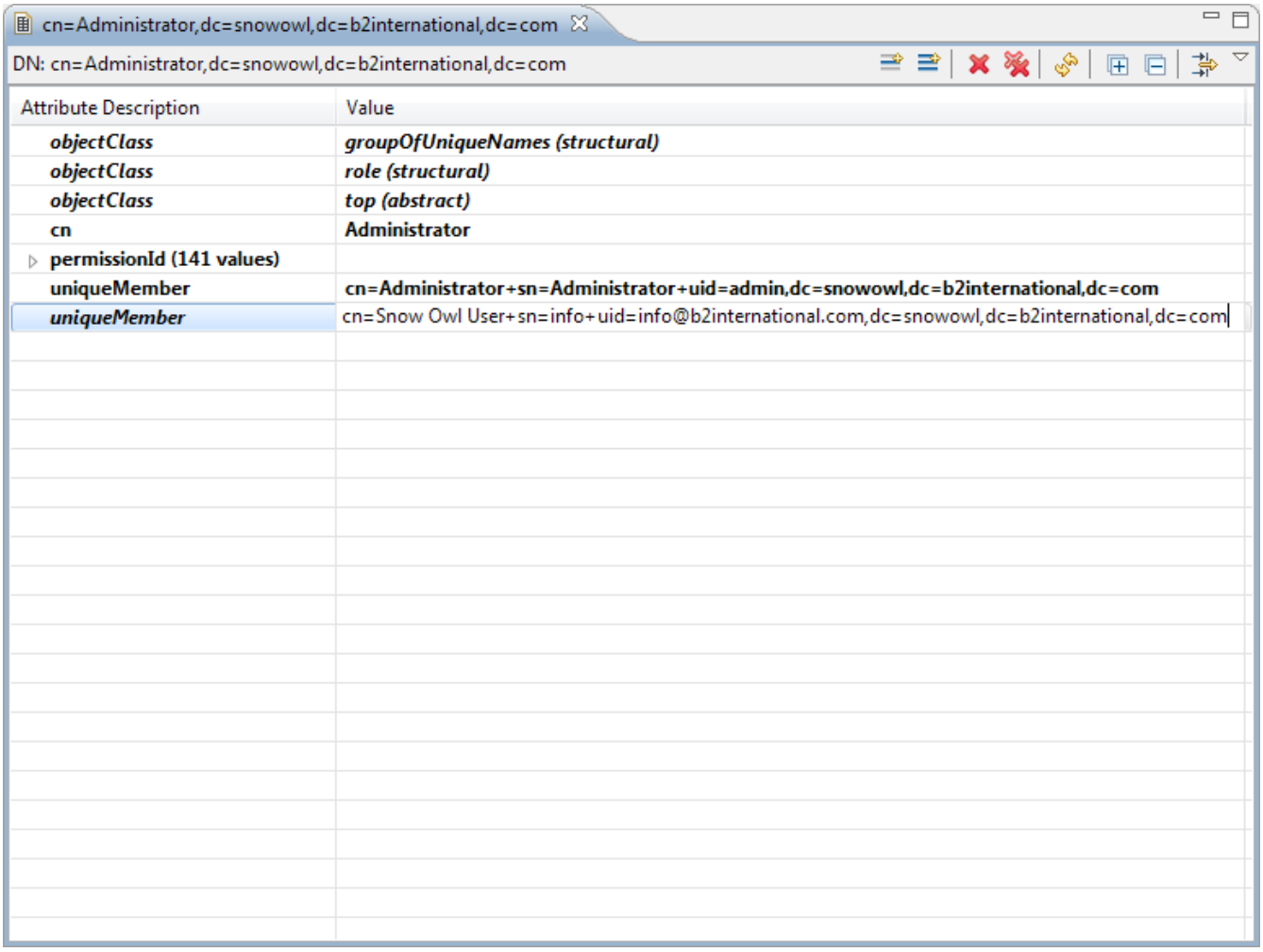


Finally, add a `uniqueMember` attribute to the Administrator group.

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:



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

Task Tracking

Finishing the installation

Create a MySQL user for Bugzilla using the command-line client:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, LOCK TABLES,
       CREATE TEMPORARY TABLES, DROP, REFERENCES ON bugs.*
       TO bugs@localhost IDENTIFIED BY 'bugzilla_pwd'; ①

mysql> FLUSH PRIVILEGES;
```

① Replace `bugzilla_pwd` with a generated password

Return to folder `/var/www/html/bugzilla` and edit `localconfig` to reflect the DB user name and password changes:

/var/www/html/bugzilla/localconfig

```
# Enter your database password here. It's normally advisable to specify
# a password for your bugzilla database user.
# If you use apostrophe (') or a backslash (\) in your password, you'll
# need to escape it by preceding it with a '\' character. (\') or (\)
# (Far simpler just not to use those characters.)
$db_pass = 'bugzilla_pwd';
```

Apply the following patch on `/var/www/html/bugzilla/Bugzilla/DB/Mysql.pm` to make Bugzilla work with MySQL 5.6:

Mysql.pm.patch

```
--- Mysql.pm.old        2015-07-23 22:07:27.797000043 +0200
+++ Mysql.pm    2015-07-23 22:10:49.373999897 +0200
@@ -309,8 +309,8 @@
     # works if InnoDB is off. (Particularly if we've already converted the
     # tables to InnoDB.)
     my ($innodb_on) = @{$self->selectcol_arrayref(
-        q{SHOW VARIABLES LIKE '%have_innodb%'}, {Columns=>[2]})};
-    if ($innodb_on ne 'YES') {
+        q{SHOW ENGINES}, {Columns=>[2]})};
+    if ($innodb_on ne 'YES' && $innodb_on ne 'DEFAULT') {
         print <<EOT;
 InnoDB is disabled in your MySQL installation.
 Bugzilla requires InnoDB to be enabled.
```

Finally, run `./checksetup.pl` again. Bugzilla should be reachable at http://localhost/bugzilla after configuration is completed. Details of the administrator user will be requested at the end of the process:

```
...
Adding a new user setting called 'per_bug_queries'
Adding a new user setting called 'zoom_textareas'
Adding a new user setting called 'csv_colsepchar'
Adding a new user setting called 'state_addselfcc'
Adding a new user setting called 'comment_sort_order'
Adding a new user setting called 'display_quips'

Looks like we don't have an administrator set up yet. Either this is
your first time using Bugzilla, or your administrator's privileges
might have accidentally been deleted.

Enter the e-mail address of the administrator: info@b2international.com ①
Enter the real name of the administrator: Administrator ②
Enter a password for the administrator account: ③
Please retype the password to verify:
info@b2international.com is now set up as an administrator.
Creating initial dummy product 'TestProduct'...

Now that you have installed Bugzilla, you should visit the 'Parameters'
page (linked in the footer of the Administrator account) to ensure it
is set up as you wish - this includes setting the 'urlbase' option to
the correct URL.
```

① Enter the e-mail address of the administrator user

② Add a display name for the Bugzilla administrator

③ Enter the password of the administrator user

Once Bugzilla has created its table structure, you can increase the maximum table size by executing the following commands:

```
mysql> USE bugzilla
mysql> ALTER TABLE attachments
       AVG_ROW_LENGTH=1000000, MAX_ROWS=20000;
```

Administration of Bugzilla

See http://www.bugzilla.org/docs/3.6/en/html/administration.html for a comprehensive list of administrative tasks and options.

After logging in with an account that has administrative privileges, click the `Administration` link on the top. The general administrative page will appear as shown below:

Core parameters can be set by selecting `Parameters` on the top left. The following fields are recommended to be adjusted:

Required Settings

urlbase

Set to the the common leading part of all URLs which are related to Bugzilla (ex.: http://server.domain/bugzilla/)

cookiepath

The common path segment of the URL under which Bugzilla cookies are allowed to be read; as noted in the description above the field, its value should begin with '/' (ex.: /bugzilla/)

General

maintainer

The email address entered here is shown on various pages in Bugzilla where contacting the administrator is suggested

User Authentication

requirelogin

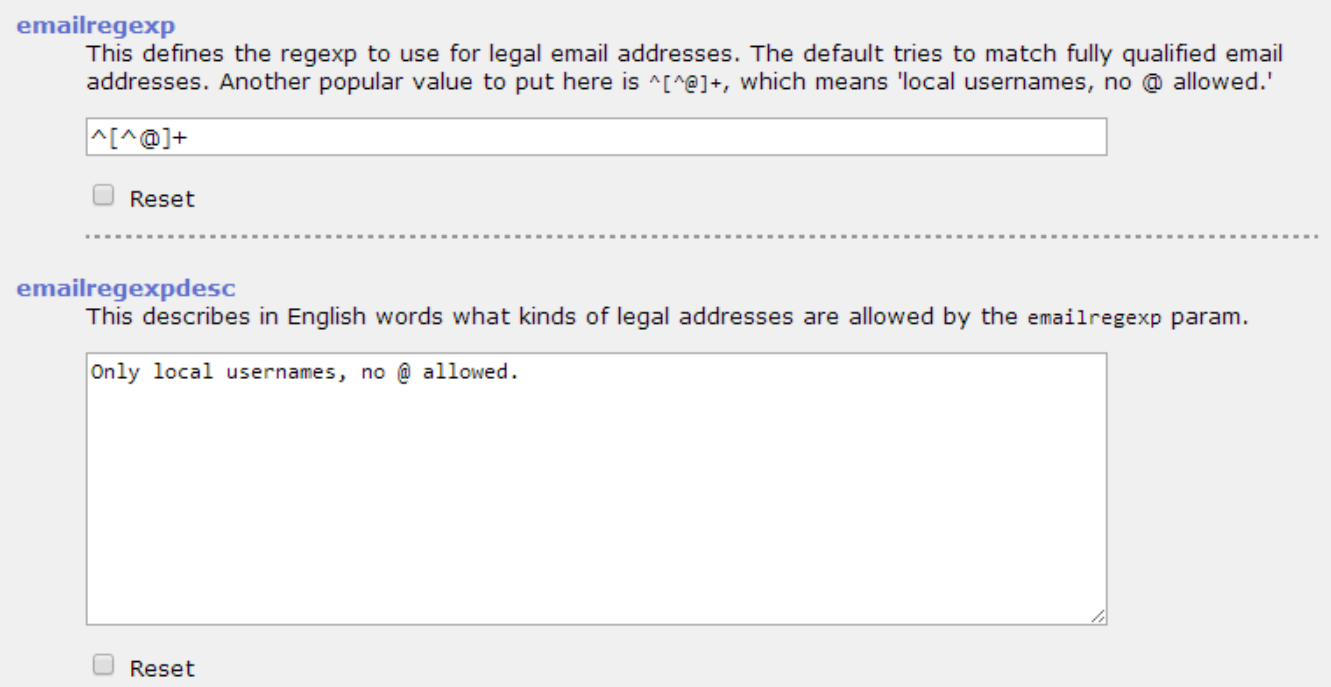
Set to On if you want to limit access to registered users only (disabling anonymous browsing of bugs)

emailregexp and emailregexpdesc

Depending on requirements, the administrator may limit login names to values that are not actual email addresses. In this case, set the fields as suggested in the description above, ie. `^[^@]+` and `Local usernames, no @ allowed.`

createemailregexp

to disable user-initiated registration (requiring the administrator to create each user account by hand), clear the field's contents



Attachments

maxattachmentsize

The maximum size in kilobytes for attachments. Change it to `10240` (10 MB)

Dependency Graphs

webdotbase

To disable relying on an external service for rendering dependency graphs of issues (as populated by default), clear the field's contents

Email

mail_delivery_method

If an SMTP server is available, configure its address and authentication properties below; otherwise, set this value to `None` to disable sending mail altogether

smtpserver

Clear the field's contents if no SMTP server is used

whinedays

Set to 0 if mail delivery is not enabled and/or there's no need to send users regular notifications about their assigned bugs which remained in NEW state

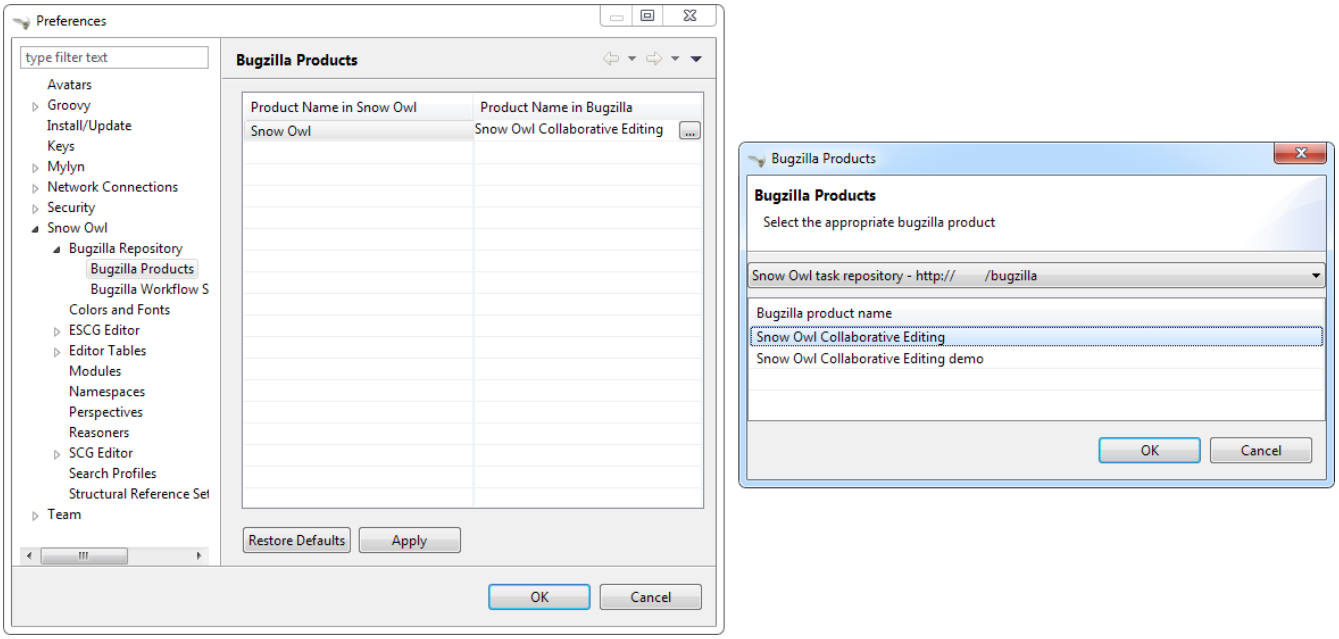
use-mailer-queue

When set to `On`, e-mails are sent asynchronously; to use this feature, the jobqueue.pl daemon needs to be started. For more information on this topic, please see http://www.bugzilla.org/docs/3.6/en/html/api/jobqueue.html.

Product setup

Bugzilla tracks the authoring aspects of Snow Owl clients in multiple products. Per-product configuration is shown in the following parts of the guide.

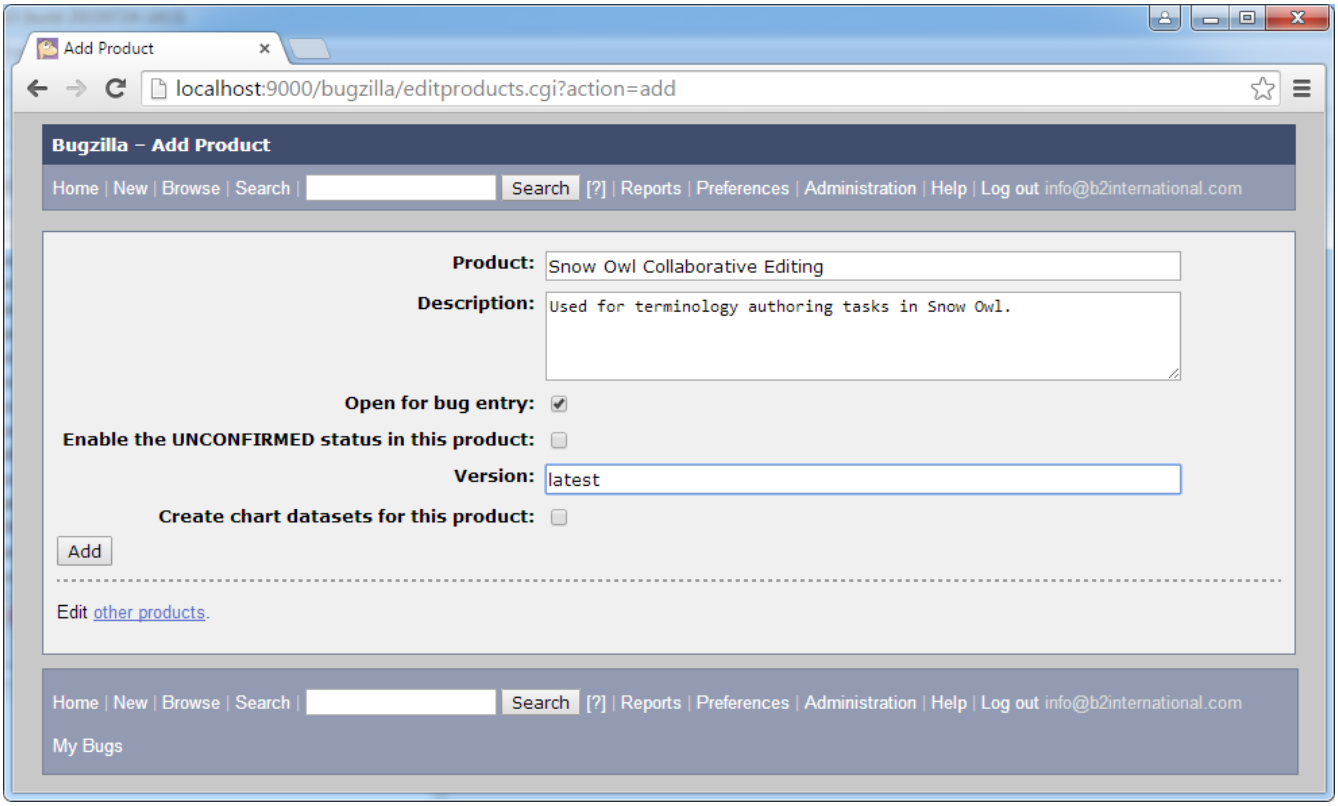
Opening the preference page `Snow Owl > Bugzilla Products` displays the supported products in the client and their corresponding product names in Bugzilla. If you have different product names added in the issue tracker, you have to adjust the product name as shown in the image. Make sure to press Enter or click in the table to apply the change in the field before hitting `Apply` or `OK` to apply the changes. Products which are not handled by contributed task editors are displayed with an empty context view only:

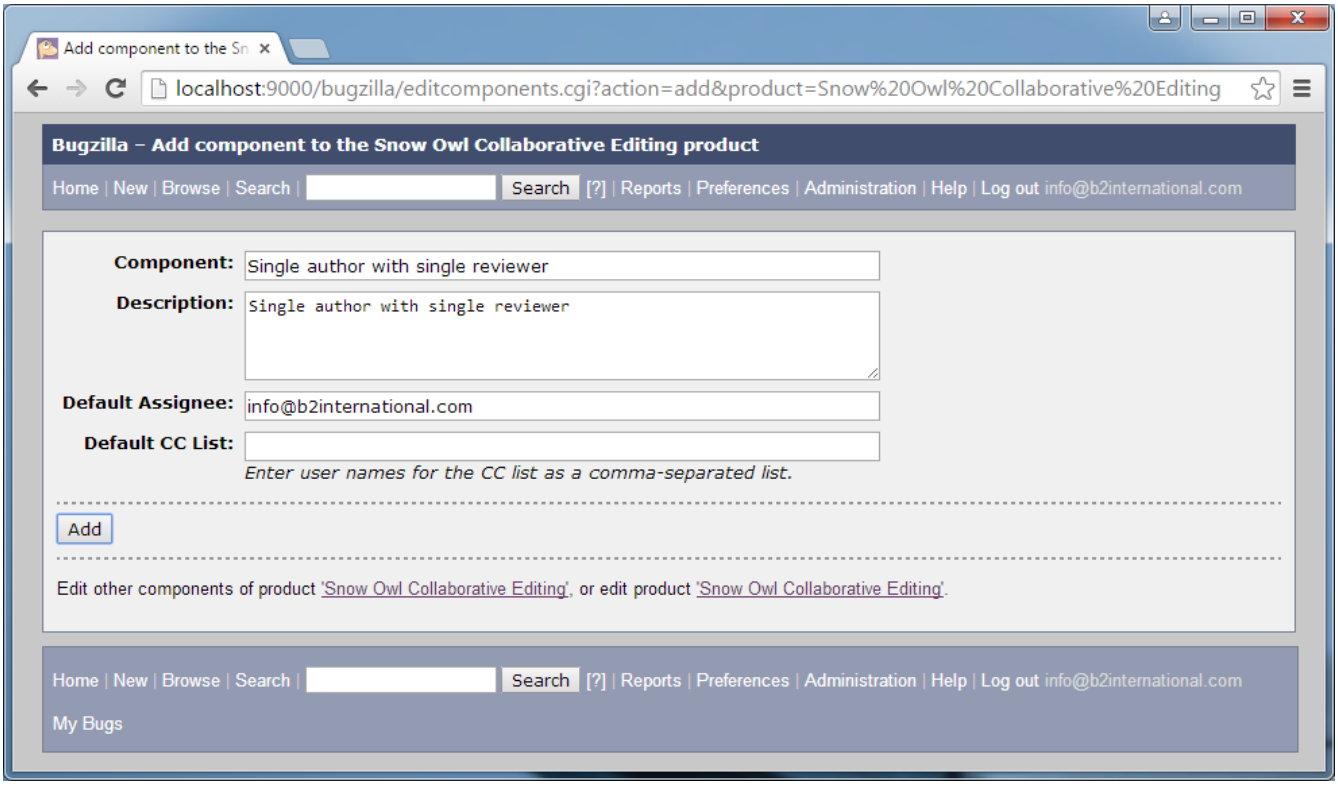


To match the default value set in client preferences, create a product called `Snow Owl Collaborative Editing` by clicking `Products` on Bugzilla's administration page. Add a description, optionally set a version to discern individual releases, and keep `Open for bug entry` checked to allow users to file

issues under this product. After creating the product, a warning will be issued by Bugzilla to create a component as well. Add the following components with the `Component`, `Component description` and `Default assignee` fields populated:

Component name	Description
Single author with single reviewer	Single author with single reviewer
Dual authors with single reviewer – Dual authoring	Dual authors with single reviewer – Dual authoring
Dual authors with single reviewer – Dual blind authoring	Dual authors with single reviewer – Dual blind authoring
Dual authors with dual reviewers – Dual authoring	Dual authors with dual reviewers – Dual authoring
Dual authors with dual reviewers – Dual blind authoring	Dual authors with dual reviewers – Dual blind authoring





Add custom fields through the web interface (`Administration > Custom fields`):

Field name	Description	Sortkey	Type	Editable on Bug Creation	In Bugmail on Bug Creation
cf_artifacttype	Task artifact type	400	Free Text	true	false
cf_author_one	Author one	410	Free Text	true	true
cf_author_two	Author two	420	Free Text	true	true
cf_reviewer_one	Reviewer one	430	Free Text	true	true
cf_reviewer_two	Reviewer two	440	Free Text	true	true
cf_adjudicator	Adjudicator	450	Free Text	true	true
cf_artifact_properties_source	Properties source	991	Free Text	true	false
cf_mappingset_id	Mapping set ID	992	Free Text	true	false
cf_valueset_id	Value domain ID	993	Free Text	true	false
cf_is_promoted	Promoted	995	Free Text	true	false
cf_parent_refset_map_target_component_type	Reference set map target component type	996	Free Text	true	false

Field name	Description	Sortkey	Type	Editable on Bug Creation	In Bugmail on Bug Creation
cf_parent_refset_referenced_component_type	Reference set referenced component type	997	Free Text	true	false
cf_parent_refset_identifierconcept_id	Parent reference set identifier concept id	998	Free Text	true	false
cf_refset_identifierconcept_id	Reference set identifier concept	999	Free Text	true	false

Authentication against LDAP

Bugzilla is capable to authenticate against the external LDAP server which Snow Owl Server will use. Setting it up requires the following steps to be taken:

- Go to `Administration > Parameters > LDAP` and populate the following fields:

  LDAPserver
  : hostname:port pair for contacting the server, eg. `localhost:10389`

  LDAPBaseDN
  : set to `dc=snowowl,dc=b2international,dc=com`

  LDAPuidattribute
  : set to `uid`

  LDAPmailattribute
  : set to `uid`

- Click `Save Changes` to apply changes
- Go to `Administration > Parameters > User Authentication`, scroll down to `user_verify_class` and make `LDAP` the top-most item
- Click `Save Changes` to finish

To test, click the `Log Out` link at the top and try to log in with your bugzilla username and LDAP password. If it was sucessful, you should see bugzilla's landing page. If you see an error message about not able to connect to the LDAP server, then run the following command as root:

```
# setsebool -P httpd_can_network_connect on
```

This will allow Apache to make network connections.

If LDAP is still not working and you are being locked out from bugzilla, you can change back bugzilla to use its internal database for authentication, instead of LDAP. To do so, edit `/var/www/html/bugzilla/data/params`, deleting LDAP from the `user_verify_class` entry:

/var/www/html/bugzilla/data/params

```
...
'user_verify_class' => 'DB',
...
```

If users are already entered in the LDAP server, it is important to synchronize Bugzilla's user database to contents of LDAP so tasks can be assigned to all users. Run the following script to perform synchronization:

```
# cd /var/www/html/bugzilla
# ./contrib/syncLDAP.pl
```

For general questions and documentation, please refer to chapter 3.1.10. LDAP Authentication in the documentation: http://www.bugzilla.org/docs/3.6/en/html/parameters.html.

Snow Owl Server

DB connection

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

```
Unresolved directive in configuration_guide.adoc -
include::scripts/snowowl_create_db.sh[]
```

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

snowowl_load_db.sh

```
Unresolved directive in configuration_guide.adoc -
include::scripts/snowowl_load_db.sh[]
```

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

```
repository:
  ...

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

① Update MySQL username and password, if neccessary

File Authentication

To use file based authentication, configure the `auth` property to `PROP_FILE` and the `fileAuth` configuration object to the desired username/password in the `snowowl_config.yml` file.

WARNING	Building Snow Owl produces a ready to use, deployable package with the default passwords (`fileAuth` and `database` properties) configured. Usage of the default password configuration is not recommended in production environments.

LDAP Authentication

To use the configured LDAP instance, adjust `snowowl_jaas_configuration.properties` to include the host name of the LDAP server and the LDAP administrator password:

/opt/snowowl-community_{version}/configuration/snowowl_jaas_configuration.properties

```
LDAP {
    org.eclipse.equinox.security.auth.module.ExtensionLoginModule required
        extensionId="com.b2international.snowowl.authentication.ldap.ldapLoginModule"
        ...
        userProvider="ldap://<ldap_host>:10389/" ①
        usePool=false
        snowOwlBase="dc=snowowl,dc=b2international,dc=com"
        bindDnUser="uid=admin,ou=system"
        bindDnPassword="secret" ②
        ...
};
```

① Replace `<ldap_host>` with the host name of the LDAP server to connect to

② Replace `secret` with the LDAP administrator's password

Memory settings

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

```
JAVA_OPTS="$JAVA_OPTS \
    -Xms8g \
    -Xmx10g \
    -XX:MaxPermSize=512m \
```

`Xms` sets the minimum heap size, `Xmx` sets the maximum heap size, and `XX:MaxPermsize` sets the PermGen space used by the JVM.

OSGi console

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

/opt/snowowl-community_{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-3.6.1.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>
```

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-community_{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-community_{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-3.6.1.RELEASE/docs/virgo-user-guide/html/ch09.html and http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.0.M04/docs/virgo-user-guide/html/ch13s06.html#configuring-authentication.

Configuration reference

Snow Owl Server comes with a predefined default configuration file, which can be used to tweak various system parameters. The configuration file is in YAML format, and located at `/opt/snowowl-community_{version}/snowowl_config.yml`. You can read more about how to create/write such files here: http://en.wikipedia.org/wiki/YAML.

The configuration file has a hierarchical structure, which is defined by modules. Different modules can have different configurations, a module is defined by its name, which should start a line in the file. Configuration parameters in a module should be indented by two spaces following the module's name.

The next section contains the reference of our currently supported configuration parameters. Each parameter should be present in a module configuration as described above.

Snow Owl Server refuses to start if the configuration file contains syntactical or structural errors. The cause of the problem can be found in the `log.log` file, or in the console if you've redirected the output of the server's startup process.

Authentication

Name	Default	Description
type	`LDAP`	`PROP_FILE`, `LDAP` - choose which type of authentication method you want. NOTE: `PROP_FILE` is supported on standalone environments only.

```
authentication:
  type: PROP_FILE
```

Repository

Name	Default	Description
host	`0.0.0.0`	The host name to bind to.
port	`2036`	The port of the chosen network interface to use when listening for connections.
numberOfWorkers	`3 x NumberOfCores`	The number of worker threads to assign to a repository during initialization.
revisionCache	`true`	Enable CDO revision cache to keep data returned from the database.
readerPoolCapacity	`7`	The capacity of the reader pool associated with the SNOMED CT store.
writerPoolCapacity	`3`	The capacity of the writer pool associated with the SNOMED CT store.

```
repository:
  host: 0.0.0.0
  port: 2036
```

Database

Name	Default	Description
directory	`store`	The directory of the embedded database inside the global resources folder where the application should look for the database files by default (if no location parameter is given).
type	`h2`	The type of the database adapter to use when connecting to the database.
driverClass	`org.h2.Driver`	The fully qualified name of the driver's Java class to use when connecting to the database.
datasourceClass	`org.h2.jdbcx.JdbcDataSource`	The fully qualified name of the datasource's Java class to use when connecting to the database.
scheme	`jdbc:h2:`	The scheme to use when connecting to the database.
location		The location of the database when connecting to it. If not set then in embedded mode the default directory parameter will be used as location.
username		The username of the database user to use when connecting to the database.
password		The password of the database user to use when connecting to the database.
settings		Other database specific JDBC settings to use when connecting to the database.

```
repository:
  database:
    directory: store
    type: h2
    username: admin
    password: admin
```

Index

Name	Default	Description
commitInterval	15000	The hard commit interval of an index in milliseconds.
translogSyncInterval	5000	The sync interval of the transaction log in milliseconds.
queryWarnThreshold	400	The threshold of the warn log when querying data.
queryInfoThreshold	300	The threshold of the info log when querying data.
queryDebugThreshold	100	The threshold of the debug log when querying data.
queryTraceThreshold	50	The threshold of the trace log when querying data.
fetchWarnThreshold	200	The threshold of the warn log when fetching data.
fetchInfoThreshold	100	The threshold of the info log when fetching data.
fetchDebugThreshold	50	The threshold of the debug log when fetching data.
fetchTraceThreshold	10	The threshold of the trace log when fetching data.

```
repository:
  index:
    commitInterval: 5000
    translogSyncInterval: 1000
    queryWarnThreshold: 400
    fetchInfoThreshold: 100
```

RPC

RPC is a custom protocol implementation used to solve request-response based communication between a client and a server.

NOTE	Changing these settings is not recommended and currently unsupported in production environments.

Name	Default	Description
logging	`false`	`true`, `false`, `ON`, `OFF` - enable or disable verbose logging during RPC communication
compressed	`false`	`true`, `false`, `ON`, `OFF` - enable or disable GZIP compression of the communication.

```
rpc:
  logging: true
  compressed: false
```

Metrics

Snow Owl can measure and report execution times (and other metrics in the future) of executed requests.

Name	Default	Description
enabled	`true`	`true`, `false`, `ON`, `OFF` - enable or disable metrics in the application

SNOMED CT

Configuration of SNOMED CT terminology services.

Name	Default	Description
readerPoolCapacity	`7`	The capacity of the reader pool associated with the SNOMED CT store.
writerPoolCapacity	`3`	The capacity of the writer pool associated with the SNOMED CT store.
language	`en-gb`	`en-gb`, `en-us`, `en-sg` - The language code to use for SNOMED CT Descriptions. Descriptions with membership of the chosen language's reference set will be used runtime.

Name	Default	Description
maxReasonerCount	2	The maximum number of reasoners permitted to do computation simultaneously. Minimum 1, maximum 3 is allowed. If the value is set to 1, classification requests will be processed in a sequential fashion.
maxReasonerResults	10	The number of inferred taxonomies that should be kept in memory after the reasoner completes the computational stage. The user can only choose to save the results of the classification run if the corresponding taxonomy instance is still present.
maxReasonerRuns	1000	The number of classification runs of which details should be preserved on disk. Details include inferred and redundant relationships, the list of equivalent concepts found during classification, and classification run metadata (start and end times, status, requesting user, reasoner used for this run).
showReasonerUsageWarning	true	'true' will display a dialog if any user selects a non-ELK reasoner, citing memory and compatibility problems, also recommending to contact B2i.
concreteDomainSupport	false	'true' will turn on support for concrete domains.
inferredEditingEnabled	false	'true' will enable manual editing of inferred relationships and concrete domain elements.

```
snomed:
  language: en-gb
  maxReasonerCount: 1
  maxReasonerResults: 20
  showReasonerUsageWarning: true
  concreteDomainSupport: true
  inferredEditingEnabled: false
```

SNOMED CT Component Identifier Configuration

Snow Owl's SNOMED CT identifier service an be configured to be either using the built-in or IHTSDO's external Component Identifier Service (CIS). The configuration needs to be placed within the **snomed/ids** section. If omitted, then default configuration will be used, which is the built-in (embedded) service based on the index store allocating ids in a sequential fashion.

Name	Default	Description
service	EMBEDDED	`EMBEDDED` or `CIS` - The service used to generate ids.
source	INDEX	`INDEX` or `MEMORY` - The source of the generated ids. `MEMORY` is used for testing.
strategy	SEQUENTIAL	`SEQUENTIAL` or `RANDOM` - The strategy of the id generation.
cisBaseUrl		The service's URL with port and without context root.
cisContextRoot		The context root of the id generation service.
cisUserName		The registered user name at the CIS site.
cisPassword		The password for the registered user name at the CIS site.
cisClientSoftwareKey	Snow Owl	The client software key to be persisted within CIS as reference.
cisNumberOfPollTries	1	The maximum number of tries when polling jobs of bulk requests.
cisTimeBetweenPollTries	1000	The time to wait between 2 job polling actions It is in milliseconds.
cisNumberOfReauthTries	2	The maximum number of re-authentication attempts when a 401 Not authorized response is received.
cisMaxConnections	100	Maximum number of simultaneous connections that Snow Owl can make to the CIS host via HTTP.

Name	Default	Description
maxIdGenerationAttempts	1000	Maximum number of attempts any non-CIS ID generator will take to generate a single SNOMED CT identifier, if exceeded it throws an exception.

Example for using the built-in service with random ids using the index as the source:

```
snomed:
  ...
  ids:
    service: EMBEDDED
    source: INDEX
    strategy : RANDOM
    cisBaseUrl : <cis_host_and_port>
    cisContextRoot : api
    cisUserName : <your-cis-username>
    cisPassword : <your-cis-password>
    cisClientSoftwareKey : Snow Owl dev. deployment
    cisNumberOfPollTries : 1
    cisTimeBetweenPollTries : 1000
    cisMaxConnections: 100
  ...
```

Example for using IHTSDO's external CIS service:

```
snomed:
  ...
  ids:
    service : CIS
    cisBaseUrl : <cis_host_and_port>
    cisContextRoot : api
    cisUserName : <your-cis-username>
    cisPassword : <your-cis-password>
    cisClientSoftwareKey : Snow Owl dev. deployment
    cisNumberOfPollTries : 1
    cisTimeBetweenPollTries : 1000
    cisMaxConnections: 100
  ...
```

Logging

Log files are stored under `./opt/snowowl-community_{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. In case the file reaches a pre-defined maximum size, the system will create additional files named `log_1.log`, `log_2.log`, etc. 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-3.6.1.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.FixedWindowRollingPolicy">
        <FileNamePattern>serviceability/logs/log_%i.log</FileNamePattern>
        <MinIndex>1</MinIndex>
        <MaxIndex>4</MaxIndex>
    </rollingPolicy>
    <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
        <MaxFileSize>10MB</MaxFileSize>
    </triggeringPolicy>
    <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.

Virgo documentation

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