

# Repo Ninja



Repo ninja command line tool was removed from MidPoint 3.7. It's replaced by new tool called just ninja. See [this](#) for details.

Repo Ninja is a tool designated to help MidPoint administrators with repository initialization and validation. It also allows to export and import repository data as XML objects even when the application is not running.

## Running the Repo Ninja: Binary Distribution

The midPoint binary distribution packages (`midpoint-X.X-dist.zip`, `midpoint-X.X-dist.tar.gz`, `midpoint-X.X-dist.tar.bz2`) contain the binary distribution of "Repo Ninja" utility. Just unzip/untar the file and edit the `"bin/repo-ninja"` (`"bin/repo-ninja.bat"`) script and fix the `"MIDPOINT_HOME"` variable to point to your `"midpoint.home"` location. Save the script and run it.

### NOTE:

When extracting `repo-ninja-3.1-SNAPSHOT-bin.zip` file in `*/midpoint-home/bin`, **check `repo-ninja.bat` file**, which content has to be following (please set your own path for `midpoint-home`):

```
rem Change the following line!
set MIDPOINT_HOME="c:\midpoint-home"

rem Sets the basedir to the directory where libraries are located
set BASEDIR=%~dp0..

java -classpath "%BASEDIR%\lib\*" -Dmidpoint.home=%MIDPOINT_HOME% com.evolveum.midpoint.tools.ninja.Main %*
```

and also **check `repo-ninja` file** which content should be following:

```
#!/bin/bash

MIDPOINT_HOME=/var/opt/midpoint

BASEDIR=$(dirname "$0"/..)

exec java -classpath "$BASEDIR/lib/*" -Dmidpoint.home=$MIDPOINT_HOME com.evolveum.midpoint.tools.ninja.Main $*
```

Also **verify up to date `repo-ninja` driver version** in `*/midpoint-home/lib` such as `repo-ninja-3.1-SNAPSHOT.jar` (**has to be only one version in lib file because of mismatch risk**).

The script is provided mainly for restoring your previously-backed up repository (**execute script when tomcat is turned off**):

### Running Repo Ninja

```
Linux:
./repo-ninja -i /var/tmp/backup.xml

Windows:
*\midpoint-home\bin>repo-ninja.bat -i *\Downloads\export.xml
```

## Running the Repo Ninja: Sources

If for whatever reason you don't want to run Repo Ninja from distribution package, you can also run it as a runnable JAR archive. After build from sources the jar is located in `tools/repo-ninja/target/release/repo-ninja-*.jar`. Use a command similar to the following to run it:

```
java -jar tools/repo-ninja/target/release/repo-ninja-2.2-SNAPSHOT.jar
```

The `lib` directory with all of its content is also needed to run the ninja. If you are transferring ninja to another machine be sure to get the main `jar` and also the `lib` directory and all the jars inside.

**Tip:** there is a execution script located in `src/other/ninja.sh`



### Experimental

Repo Ninja is still an experimental tool. It needs a lot of knowledge about the system and lot of trickery to use. Use it carefully.

## Repo Ninja Usages

The following options and examples document the usage of Repo Ninja.

### Repository validation

Use options:

```
-v
-Dmidpoint.home=<midpoint configuration folder>
```

`<hibernateHbm2ddl>` option in `config.xml` in MidPoint configuration should be set to **validate**.

Ninja tries to load spring contexts with Repository service. This operation finishes with SUCCESS (if connection to database and database schema looks right) or FAILURE.

### SQL script execution

Use options:

```
-c <file> to execute create database schema DDL script which you can find on [midPoint:Repository
Configuration] page.
-d <driver class name>
-P <jdbc password prompt>
-p <jdbc password>
-U <jdbc username>
-u <jdbc url>
```

### Export midPoint objects to XML file (Create backup)

```
-e <file>
-Dmidpoint.home=<midpoint configuration folder>
```

### Import midPoint objects from XML file (Restore from backup)

```
-i <file>
-Dmidpoint.home=<midpoint configuration folder>
```

## Known problems

1. Repository validation fails when MidPoint configuration uses `<dataSource>` and doesn't contain simple JDBC connection parameters. It's because JNDI data source is not available outside tomcat container so Repo Ninja can't see it and therefore can't initialize Repository service.
2. `-Dmidpoint.home=<midpoint configuration folder>` has to be before `-jar` option so that JVM finds it as system property. The binary distribution scripts set this property for all invocations.
3. Escape semicolon `'` in option `jdbc url`, for example:

```
-u jdbc:sqlserver://192.168.1.27:1433\;database=midpoint
```

4. JDBC drivers (mysql, postgresql, h2) are already bundled. If you use Ninja with Oracle or MS SQL Server you have to add jdbc driver jar libraries to `lib` folder before you run this application.