

# Command Utilities

This section details the various command utilities that are available with OpenDNSSEC.

## On this Page

- [ods-control](#)
- [ods-ksmutil](#)
- [ods-signer](#)
- [ods-hsmutil](#)
- [ods-getconf](#)
- [ods-auditor](#)
- [ods-hsmspeed](#)
- [ods-kaspcheck](#)
- [hsmbully](#)
- [Daemons](#)
  - [ods-signerd](#)
  - [ods-enforcerd](#)

## ods-control

Is a wrapper around the commands below.

```
usage: ods-control ksm|hsm|signer|start|stop
```

- The first three options pipe commands to *ods-ksmutil*, *ods-hsmutil*, and *ods-signer*.
- The last two options start and stop the two daemons of OpenDNSSEC, *ods-enforcerd* and *ods-signerd*.

## ods-ksmutil

You need a way to interact to the KASP Enforcer, for example to add and remove zones that are handled by OpenDNSSEC. The *ods-ksmutil* utility provides a number of commands to make this easier, all commands are invoked on the unix command line.

- **You must run the *setup* option before you ever run any sub-system in OpenDNSSEC.** This reads the configuration *kasp.xml* and imports these settings into the KASP Enforcer database.

The *setup* command deletes the current content of the database! (Including information on keys; such that existing keys will become unusable and new keys will need to be generated.)

- If you make any changes to *kasp.xml* these changes must be imported into the database. Use the *update* command to do this without losing any other data.
- To add a zone to be handled by OpenDNSSEC, use the *zone add* command. This command needs a parameter to specify the zone, and optional parameters for which policy to use and which paths to use for input and output. An example of use:

```
ods-ksmutil zone add -z example.com -p default -i /var/example.com -o /var/example.com.signed
```

- The command *zone delete* is simpler and needs no further parameters but the name of the zone.

A complete list of commands can be found by running:

```
ods-ksmutil -h
```

or they are shown in detail here: [ods-ksmutil commands](#)

## ods-signer

The *ods-signer* provides a Command Line Interface to the *ods-signerd*. There are a number of commands you give to *ods-signer*. If you start the CLI without any command line parameters you enter a shell where you can issue commands:

```

ods-signer
cmd> help
Commands:
zones                Show the currently known zones
sign <zone> [--serial <nr>] Read zone and schedule zone for immediate (re-)signing.
                        If a serial is given, that serial is used in the output zone.
sign --all            Read all zones and schedule all for immediate (re-)signing.
clear <zone>         Delete the internal storage of this zone.
                        All signatures will be regenerated on the next re-sign.
queue                Show the current task queue.
debug-locks          show locking information (for debugging purposes).
flush                Execute all scheduled tasks immediately.
update <zone>        Update this zone signer configurations.
update [--all]       Update zone list and all signer configurations.
start                Start the engine.
running              Check if the engine is running.
reload               Reload the engine.
stop                 Stop the engine.
verbosity <nr>      Set verbosity.

```

The same commands can be passed as command line arguments in your unix shell.

- The **debug-locks** utility was added in 1.3.14.
- The **sign <zone> [--serial <nr>]** option was added in 1.3.15.

## ods-hsmutil

The *ods-hsmutil* utility is designed to interact directly with your HSM and can be used to manually list, create or delete keys. It can also be used to perform a set of basics HSM tests.

Be careful before create or deleting keys using *ods-hsmutil*, as the changes are **not** synced with the KASP Enforcer.

## ods-getconf

The tool *ods-getconf* allows you to retrieve a configuration value from conf.xml, given an expression. Introduced in 1.3.18.

## ods-auditor

The Auditor (*ods-auditor*) can do an audit of the zones in the system to see if the signer complies to what the policy mandates. It is run automatically (unless disabled) after each resigning of a zone and will stop the signed zone from being distributed if it finds any issues. Any errors found by the *ods-auditor* will be logged to the configured syslog utility. This should be checked for debug if you have issues.

You can also run the Auditor yourself, to get feedback on the current status, to loop through all zones run:

```
ods-auditor
```

or, to audit just one zone, run:

```
ods-auditor -z <zone>
```

It is possible to override the audit type specified in the kasp.xml Policy for the zone. To run a full audit, use the `--full` flag, and use `--partial` to force a partial audit of the zone.

 If you are using the partial auditor to audit your very large zone, you may wish to run an occasional off-line full audit. To do this, take a copy of your signed and unsigned zone files, and run :

```
ods-auditor -z <zone> --full --signed <path/to/signed/file> --unsigned <path/to/unsigned/file>
```

## ods-hsmspeed

The tool *ods-hsmspeed* does performance testing on your HSM. This is also useful to find out at what speed you can get from SoftHSM on your CPU.

## ods-kaspcheck

This tool is provided to check that the configuration files (conf.xml and kasp.xml) are semantically sane and contain no inconsistencies.

It is advisable to use this tool to check your configuration before starting to use OpenDNSSEC.

```
ods-kaspcheck -h
Usage: ods-kaspcheck options
Specific options:
  -c, --conf PATH_TO_CONF_FILE Path to OpenDNSSEC configuration file
                                (defaults to the default conf.xml file)
  -k, --kasp PATH_TO_KASP_FILE Path to KASP policy file
                                (defaults to the path given in the configuration file)
Common options:
  -h, -?, --help                Show this message
```

## hsmbully

The hsmbully tool may be used to test your HSM for compliance with PKCS#11. This tool is not part of OpenDNSSEC, but can be found in the GitHub repository:

```
git clone https://github.com/opendnssec/hsmbully.git
```

## Daemons

You can also run the two OpenDNSSEC daemons *ods-signerd* and *ods-enforcerd* from the command line, they are installed into the sbin directory.

### ods-signerd

This is the component that performs all of the signing. It first reads zonelist.xml and then goes through all zones to sign them if needed. Start the daemon by running:

```
>ods-signer start
```

or if you want to use specific command line options:

```
>ods-signerd -h
Usage: ods-signerd [OPTIONS]
Start the OpenDNSSEC signer engine daemon.

Supported options:
-c | --config <cfgfile> Read configuration from file.
-d | --no-daemon        Do not daemonize the signer engine.
-l | --single-run       Run once, then exit.
-h | --help             Show this help and exit.
-i | --info             Print configuration and exit.
-v | --verbose          Increase verbosity.
-V | --version          Show version and exit.

BSD licensed, see LICENSE in source package for details.
Version 1.3.8. Report bugs to <http://bugs.opendnssec.org/>.
```

### ods-enforcerd

The Enforcer daemon creates keys if needed (and configured to); it also maintains the states of the keys according to the appropriate policy. As the states of keys change, it communicates these changes to the signer via the configuration files that the signer uses when signing the zones. To run, call:

```
>ods-enforcerd
```

or if you want to use specific command line options:

```
>ods-enforcerd -h

Usage: ods-enforcerd [OPTION]...
OpenDNSSEC Enforcer version x.y.z

Supported options:
-c <file>    Use alternate conf.xml.
-d           Debug.
-l           Run once, then exit.
-p <policy>  Run once processing only the specified policy, then exit.
-P <pidfile> Specify the PID file to write.
-V           Print version.
-[-?|h]     This help.
```

Note that the **-p <policy>** option is available in 1.3.16.