Administration Guides for Linux Products

Administration Guide

Next Generation Network
Access Technology

www.ncp-e.com

Contact

For more information or questions about NCP products and services:

Germany
NCP engineering GmbH
Dombühlerstraße 2
D-90449 Nürnberg
Tel.: +49 (911) 9968 0
Homepage: http://www.ncp-e.com
Mail: info@ncp-e.com

Contact USA, North American HQ
NCP engineering, Inc.
601 Cleveland Street
Suite 501-25
Clearwater, FL 33755
Phone: +1 (650) 316-6273

E-Mail Support:
support@ncp-e.com (german)
helpdesk@ncp-e.com (english)

Support Hotline:
0900 / 1 99 68 00
(only available from Germany, 80 Cent / per minute)
Our support times are from monday to friday from 08:00 am to 17:00 pm.

For a support request we need the following information:
- exact product name
- serial number
- version number
- precise description of the problem
- any error message(s)

Administration Guides for Linux Products

Table of contents

1. Preface

This document explains the installation and usage of NCP products on the Linux operating system.

1.1 Scope

This documentation applies to all of the following NCP products:

- NCP Secure Enterprise Server version 12.00 and higher
- NCP Virtual Secure Enterprise VPN Server version 12.00 and higher
- NCP Secure Enterprise HA Server version 11.00 and higher
- NCP Virtual Secure Enterprise HA Server version 12.00 and higher
- NCP Friendly Net Detection Server version 2.20 and higher
- NCP Secure Client version 5.20 and higher
- NCP Secure Enterprise Management Server version 5.30 and higher

This manual does not cover the complete use of these products, but only functions that are specific to the Linux operating system. Most notably the installation of the software and its integration into Linux.

Some products contain special features that aren't available in any other products. These cases will be pointed out where necessary. Also note that some details explained in this documentation may differ between software versions as new functions are added and errors get resolved.

To take full advantage of this document, basic knowledge of the Linux operating system and the Linux command line is required.

1.2 How to read this document

Within this document you will come across the token *<prod>* in command names or filenames. This is used as a placeholder for an individual abbreviation used for each NCP product. The following table shows the *<prod>* values for the different NCP products.

Table 1: Prefixes of NCP products

Product	**Value for** `<prod>`
NCP Secure Enterprise Server	ses
NCP Virtual Secure Enterprise VPN Server	vses
NCP Secure Enterprise HA Server	has
NCP Virtual Secure Enterprise HA Server	vhas
NCP Friendly Net Detection Server	fnd

NCP Secure Client	clnt
NCP Secure Enterprise Management Server	sem

If, for example, a program appears as `<prod>-uninstall` in this documentation, then in the case of the NCP Secure Enterprise Server, this refers to a program named `ses-uninstall`. Accordingly for the other NCP products as shown above.

All commands and examples shown in this document are supposed to be run from a Linux console. Other terms for *console* used in this document are *terminal* or *command line*. In general, for any shell command `<cmd>` you can obtain a brief usage help by entering `<cmd> --help` or `<cmd> -h`.

Most of the example output shown throughout this document is taken from the installation routine and utilities of NCP Friendly Net Detection Server. However, it mostly applies to all other supported NCP products as well.

1.3 Supported Linux Distributions

NCP products run on the following Linux distributions:

- Debian GNU/Linux
- Ubuntu
- Red Hat Enterprise Linux (oder CentOS)
- SUSE Linux Enterprise

1.4 Firewall

Please notice that all products except NCP Secure Client require incoming network connectivity. Therefore, it may be necessary to modify an existing firewall configuration or to disable the firewall at all. More information about the required ports is available in the general documentation for the specific product.

1.5 Special notes for NCP Virtual Secure Enterprise VPN Server

When using NCP Virtual Secure Enterprise VPN Server, which also contains the NCP Virtual Secure Enterprise HA Server, please note that all instructions in this document regarding the system environment (for example the supported Linux distributions) do not apply, because the products are a virtual appliance. Those contain their own operating system. For instructions on virtual appliances please read the installation guide which is in the `doc` directory of the ISO image.

Updating is done through the Debian package management system via specially provided package sources. Please consult the appropriate installation help.

Both products are based on the same mechanisms and come with the same command line tools. You can access all functions via the Web interface or the management server.

2. Migration from Older Versions

If you are migrating from versions of NCP products older than listed here 1, then some major changes will occur for you.

- A new installation routine is used and the location of files has changed. Find general information about the installation process 3 and information specific to the update from old versions 6 in this document.
- The way NCP programs are started and the integration into the Linux init system has improved. Find more information in boot up and shutdown of the product 10.
- A number of new and standardized command line utilities are now part of each NCP product. Information about that is found in the section about command line tools 14.

Note

The changes to previous versions of the NCP software are significant in some places. The updates are made automatically. However, errors may occur due to the complexity of the process and the individual environment. Please read the differences to older versions carefully to avoid possible errors.

3. Installation

3.1 Running the Installer

Each NCP Linux product is delivered as a binary installation program that ends in the `.bin` file extension.

To perform the installation you first need to copy this installation program to a suitable location on the target machine.

If the `executable bit` for the installation file is not set, you must do this before. The following list shows how to check the installer program, if it has got an `executable bit`, and how to add it if necessary.

Note

The `executable bit` is necessary for the operating system to allow running the file as a program. It can be lost when the installer program is saved in ZIP archives, downloaded from the Internet or stored on removable media.

Adding the executable bit to the installer program

```
$ ls -l fnd_linux_x86-64_200_rev16909.bin

-rw-rw-r-- ① 1 user user 27887112 30. Apr 13:50 fnd_linux_x86-64_200_rev16909.bin
$ chmod +x fnd_linux_x86-64_200_rev16909.bin
$ ls -l fnd_linux_x86-64_200_rev16909.bin

-rwxrwxr-x ② 1 user user 27887112 30. Apr 13:50 fnd_linux_x86-64_200_rev16909.bin
```

① If `x` is written here, the `executable bit` is already set (not the case in this example).

② Here the bit is set after it was added using `chmod`

Run the `executable bit` once you have made sure that it is set for the installation program.

Printing the installer program help (excerpt)

```
$ ./fnd_linux_x86-64_200_rev16909.bin -h
Usage:
        ./fnd_linux_x86-64.bin      [--restore <Pfad>] [--verify] [-k]
                                    [--tempdir <Pfad>] [-x <Pfad>] [-i] [--relaxed]
                                    [--compatibility] [-v] [-d <Pfad>] [-n] [-b]
                                    [--su] [--sudo] [--] [--version] [-h]
[...]
```

The help text can be used as a source for online documentation. It explains the parameters that can be passed to the installation program. These parameters influence how certain details of the installation are performed. In the typical case you can run the installer without adding parameters.

If you want details about the NCP product included in the installation program, pass the `--info` parameter.

Information about the installer

```
$ ./fnd_linux_x86-64_200_rev16909.bin --info
                        ---------------------------------------
                        > NCP Friendly Net Detection Server <
                        ---------------------------------------
This is an installation package for:


Product code name: fnd
Product full name: NCP Friendly Net Detection Server
Product version: 2.00
Target architecture: x86_64
Target OS: linux
Build type: opt-debug
Library type: static
Size of contained data: 2772297 bytes (2.64 MB)

Environment Information:

Detected Linux Distribution: Gentoo
Detected Linux Version: Gentoo 2.2
Detected Init System: OpenRC
Tool for gaining root privilege: su
```

By passing the `--info` parameter, the installation program prints information about the data it contains and the Linux system it has detected. Afterwards the program shuts itself down without any further action.

When the installation starts, you need sufficient permissions (*root* privileges) to perform it.

To get *root* privileges, the installation program will call the `su` or `sudo` program, which will ask you for the *root* or user password, depending on your system configuration. After entering the password, the installation program will reinvoke itself with *root* privileges.

For the `su` and `sudo` programs to work they must be correctly configured. Each Linux distribution uses different default options. For the typical Linux distribution and most common cases the tool chosen by the NCP installer will be correct. In some cases, it may be necessary for you to explicitly choose the tool. Do this by passing the `--su` or `--sudo` switch as parameter to the installation program.

Following is an example of gaining root privileges upon starting the NCP installer

```
                              ------------------------------------
                              > NCP Friendly Net Detection Server <
                              ------------------------------------
Unpacking installation data... succeeded
To continue installation, root privileges are required.
The 'su' ① utility will now be called to reinvoke the installation script with ele
Please provide the required credentials
Password: ******
=== Calling installation routine ===
[...]
```

① Here the utility used to get *root* privilege is reported.

3.2 A Typical Installation Process

Before getting to the installation process, read Running the Installer 3.

To perform an installation simply run the installer program without any arguments. The installer will guide you through a series of steps until the NCP product is completely installed on the system.

In the first step the data contained in the installer will be unpacked into a temporary directory. Then some compatibility checks will be performed. These make sure that the Linux system is compatible with the contained software. If these steps succeed the installer prints information about the software that is about to be installed. The program then asks for confirmation, whether the installation should continue.

Note

By default the software will be installed in the directory `/opt/ncp/<prod>`. You can specify a different directory during the installation. However, to prevent data loss, the target directory must be empty. Please note that changing the directory after installation is more complex.

If any steps of the installation failed, have a look at Dealing with Installation Errors 10.

3.3 Update of Existing Installations

To update an NCP product, start the installation of the new version.

Run the installation program without arguments. The installer will guide you through a series of steps until the NCP product is fully installed on your system.

During the *Checking compatibility* installation step, the installation wizard checks whether the version already installed can be updated.

Note

In some cases, a special update sequence may be necessary via an intermediate version. In such a case, a corresponding error message is issued and the installation will be cancelled.

The files from the new software will overwrite previous versions of the files in the installation directory. Configuration files that are supposed to be edited by the user will not be overwritten, instead these files are accompanied by files ending in `.sam` which contain a sample configuration that can be reviewed for changes after an update.

Warning!

Downgrading an NCP installation to an older version or revision might not be fully supported and the installer will print a warning in such cases. If you are unsure about this you can contact NCP support for detailed information.

If necessary, also consider Updates from Older Versions That Use an Incompatible File Structure 6.

3.3.1 Updates from Older Versions That Use an Incompatible File Structure

Previous versions of NCP products used a different installation routine and file structure. While in the current versions most of the data is stored in one installation directory in the old versions the data was spread across different paths. These older versions of NCP products can also be updated to the new file structure. The update from the previous structure to the new one can only be performed from specific versions:

- NCP Secure Enterprise Server can be updated from version 8.11 to version 8.14
- NCP Secure Enterprise HA Server can be updated from version 3.04 to version 3.05
- NCP Friendly Net Detection Server can be updated from version 1.01 to version 2.00
- NCP Secure Client can be updated from version 3.25 to version 3.30
- NCP Secure Enterprise Management Server can be updated from version 3.02 to version 3.03

If the version of your NCP product is older than shown abover, it is necessary that you first upgrade to one of the versions listed here and then update to the installation of the new version.

The current installation routine will check for these conditions and only allow updates from the versions shown above. If the update is found to be compatible then the installer will show a slightly different version information.

During the installation you have the chance to determine the installation directory of the software. For previous installations of NCP products except for NCP Secure Enterprise Management Server the installation directory was set to the path `/usr/local/ncp/<prod>`. If you decide to change the installation directory, it may be necessary to adjust the configuration files of the NCP product according to the installation paths.

Selecting a new installation directory during the update of old installations

In the previous installation concept files often were placed flat into the installation directory. In the new installation the files are structured into subdirectories of the installation directory like `bin`, `sbin` and `etc`. Therefore during the update the installer needs to decide which existing files go into which subdirectory. All files the installer knows about are automatically put into the appropriate location.

If the user has added custom files to the installation, the installer will not recognize these files. In this case, the installation routine places them in a safe location under the `old` subdirectory. The installer also issues a warning message during the upgrade. You must put the relevant files in the desired directory yourself.

Update of old NCP Secure Enterprise Server installations that have the NCP Secure Enterprise HA Server installed

If you have already installed the NCP Secure Enterprise Server and NCP Secure Enterprise HA Server and want to update both, you must observe the following procedure. NCP Secure Enterprise HA Server can only be installed if NCP Secure Enterprise Server is already installed. To update to the new installation routine, it is necessary to first update NCP Secure Enterprise HA Server to the current version. Only then update NCP Secure Enterprise Server. The installation routine will inform you if the correct update order is not maintained and refuse an update in that case.

3.4 Product Specific Installation Features

Some NCP products have special functions during installation. These cases are covered in this section.

3.4.1 NCP Secure Enterprise Management Server

In the case of the NCP Secure Enterprise Management Server, the installer will not ask you during the installation whether you want to start the software directly. For this, a database connection 25 must be configured before it can be successfully started.

3.4.2 NCP Secure Client

For the NCP Secure Client, a group account with the name `ncp-` is created by default during installation. Only users who are members of this group can successfully use the graphical monitor application of the VPN client. The installation routine will not add any users to this group automatically. Refer to your Linux

system documentation to learn how to add a user to the ncp group. You can find out how to add a user to the *ncp* group in the documentation of your Linux system. The program `clnt-monitor` is used to start the graphical user interface.

3.5 User and Group Accounts

Some of the NCP products like NCP Secure Enterprise Server and NCP Secure Client create dedicated user and group accounts for running the software. These are necessary for user privilege seperation. This allows some services to act as a normal user without *root* privileges. This reduces the impact of possible security vulnerabilities.

By default the *ncp* user and group are used for this purpose. You may also specify a custom user and group name by passing the
`--user` and `--group` switches to the installer program. The user and/or group you specify need to exist before doing so. To create a user or group account you can use the usual Linux administration tools. Refer to your Linux system's documentation to learn more about this.

If multiple NCP products are installed on the same Linux machine, the name of the group used to run the programs must be the same for all products. This is necessary because NCP programs access shared files such as the `/etc/ncp.db` file.

3.6 Uninstallation

Warning!

After uninstalling, no backup copy remains. Please safe your data beforehand.

To uninstallt a separate program `<prod>-uninstall` exists for each NCP product. The uninstall program will inform you about the product you wish to uninstall and the affected paths. Finally, confirm whether you really want to continue with the uninstallation process.

The uninstaller will remove all product-specific files. Various system settings that have been customized for the software, such as the Linux init system, group accounts, etc. will also be reset. The only data that will remain is `/etc/ncp.db`. It is shared with other potential installations of NCP products and contains NCP product licenses you may have registered.

If you want to remove an NCP product automatically without an interactive query, you can pass the `--force` option to the program `<prod>-uninstall`.

Example output of the program fnd-uninstall

```
This program will remove the following product from your system. This includes your

        Product code name: fnd
        Product full name: NCP Friendly Net Detection Server
        Product version: 2.00
        Target architecture: x86_64
        Target OS: linux
```

```
        Build type: debug
        Library type: shared

The following paths will be removed:

- /opt/ncp/fnd
- /var/adm/ncp/fnd
- /var/log/ncp/fnd

Do you really want to perform the uninstallation?

        (yes/y/no/n): y

Removing init system integration... succeeded
Removing system PATH settings... succeeded
Removing installation files... succeeded
Purging global product configuration... succeeded
```

3.7 List of File Locations

Apart from the main installation directory, NCP software uses several other paths in the Linux file system to store data:

Location	Description
`/etc/ncp.db`	A database file that is shared between all installations of NCP software on the system. It contains some software settings and the licensing and version information for each installed product.
`/etc/ncp.info`	A configuration file where all installations of NCP products, their installation directories and versions are recorded.
`/var/log/ncp/<prod>`	Log files created by NCP programs will be stored in a dedicated directory for each product in this location.
`/var/adm/ncp/<prod>/crashes`	Information about program crashes will be collected here.

3.8 Automatic Installation

You can automatically install an NCP product without any user interaction. This is useful for quick testing or if you want to roll out NCP software via scripts.

To enable the automatic installation mode, add the basic `--batch` parameter to the installation routine. In this mode any interactive questions will be implied to be answered positively. For any configuration values that would have been queried from the user, default values are selected.

An automatic installation can only be performed by the *root* user, since the password for increasing rights cannot be read in without user interaction.

3.9 Dealing with Installation Errors

Problems may occur when trying to install or update an NCP product. There are a number of things you can try before contacting NCP support. The installation program offers some switches that help to investigate or fix installation errors.

First check the integrity of the data contained in the installation archive. This is done by passing the `--verify` option and causes the installer to verify itself and report success or failure. No installation steps are executed. If the verification fails, then the installation program is corrupt. In this case, obtain a correct version of the installation program and try again.

You can avoid some minor problems by passing the `--compatibility` switch to the installer. This changes the behavior of the installer in some places to be more compatible with unfamiliar Linux environments.

The `--relaxed` switch simply ignores certain categories of errors. This only applies to installation steps that are not elementary for the basic function of the software. Existing problems should then be corrected manually.

The `--verbose` switch causes the installer to print more background information. If you turn on *verbose* mode, every file that the installer installs and its destination will be displayed. This information gives an indication of why the installation may not work. The comprehensive output is also valuable for NCP support to investigate and resolve your problem.

4. Boot up and Shutdown of the Product

In this section you will learn how the boot up and shutdown of NCP products is done on Linux.

4.1 Manual Boot up and Shutdown

Each NCP product consists of one or more background processes that run in the system to provide the functionality of the respective product. Such background processes are called *daemons* in Linux.

The `<prod>-sentinel` program is responsible for starting all *daemons* that belong to an NCP product. It can be used to manually start up the software, e.g. to see if everything is running correctly, before the software is started automatically during the boot process.

As a first test, call the sentinel program with the `-f` switch so that it remains in the foreground and outputs information to the console. The *sentinel* program will boot up the complete NCP product. If an error occurs, the *sentinel* program will shut down the NCP product again and output an error. Otherwise, the program will continue to run until a shutdown request occurs.

Example run of the fnd-sentinel program from NCP Friendly Net Detection Server

```
$ fnd-sentinel -f
Setting core_pattern to '|/opt/ncp/fnd/bin/fnd-crash --dump %p;%u;%g;%s;%t;%h;%e;%%
   downstream core'
Starting Friendly Net Detection Daemon ... okay
product started
```

```
fnd-sentinel: started on Mo 12 Mai 2014 09:36:08 CEST
Listen Port 12521
Start FND Listener

① ^CReceived shutdown request.
Shutting down all daemons
Stopping Friendly Net Detection Daemon ... Stop FND Listener exited
Cleaning up system from VPN settings...

        Cleaning iptables mangle rules
        Cleaning unused shared memory segments
        Cleaning unused semaphores
        Cleaning unused message queues

fnd-sentinel: exited on Mo 12 Mai 2014 09:36:51 CEST
```

① Upon pressing *ctrl-c* the *sentinel* program will shut down the product again and exit after doing some cleanup operations.

Note

Some NCP products like NCP Secure Enterprise Management Server cannot be started successfully after installation if they have not been correctly configured first.

Find more information about the sentinel program here 14.

4.2 Boot up and Shutdown Using the Linux Init System

In Linux an *init* system is responsible for starting up services when booting. Currently a number of different init systems are in use in major Linux distributions.

Table 3: Init systems used in Linux distributions

Name	Description	Used in
SystemV	This is the classic UNIX-style init system using shell scripts and dependencies between them.	Debian up to version 7, openSUSE before version 12.3, SLES before version 12, backward compatibility in CentOS 6 and RHEL 6
Upstart	An advanced, event-based init system developed for Ubuntu Linux.	Ubuntu versions up to version 14, basic support in CentOS 6 and RHEL 6
systemd	A modern event-based init system with support for many modern Linux features, developed by the Linux community.	openSUSE starting from version 12.3, SLES starting from version 12

OpenRC	A niche init system developed by the Gentoo Linux community.	Recent Gentoo Linux

NCP software supports all these common init systems. You can set up NCP programs so that they are started during the system boot up. During the installation of NCP software, this integration is performed by default, if not selected otherwise. To change this autostart setting later use the <prod>-initconfig program 19.

Since different init systems are used in Linux, the commands to start or stop an NCP product differ. If you do not know what the correct commands are for your specific case, use `<prod>-initconfig`. Pass the `--show-start-cmd` and `--show-stop-cmd` parameters to it for printing the command to start and stop the NCP software respectively.

Each init system uses a basic script or service name to identify the different programs to be managed. For NCP software this basic name is `ncp-<prod>`. Note the example of starting NCP Friendly Net Detection Server when using the SystemV init system on Debian.

Starting and stopping NCP Friendly Net Detection Server on Debian Linux (SystemV init)

```
$ fnd-initconfig --show-start-cmd ①
/etc/init.d/ncp-fnd start

$ /etc/init.d/ncp-fnd start ②
Starting NCP Friendly Net Detection Server
Starting Friendly Net Detection Daemon ... okay

$ /etc/init.d/ncp-fnd status ③
Current operational status of NCP Friendly Net Detection Server

Friendly Net Detection Daemon

============================

Status: running since Mo 12 Mai 2014 04:07:16 CDT
Command Line: /usr/local/ncp/fnd/sbin/ncpfndd -f
Process ID: 5010
```

① This determines the command to start NCP Friendly Net Detection Server via the Debian init system.

② Use the start command to start NCP Friendly Net Detection Server

③ Outputs the current operating status of NCP Friendly Net Detection Server. **The command differs between init systems**.

If you did not choose to enable autostart of the NCP software during installation, change this setting afterwards using `<prod>-initconfig -a 1` to enable autostart or `<prod>-initconfig -a 0` to disable this respectively. If autostart is activated, the NCP software is booted during the system startup of your Linux system.

Note

You can also change the autostart setting using the mechanisms provided by your init system. For example on Debian Linux you can add NCP Friendly Net Detection Server to the autostart by calling `insserv --add ncp-fnd`. The `<prod>-initconfig` tool has the advantage that it is independent of the underlying init system.

If you have installed NCP Secure Enterprise HA Server, which depends on NCP Secure Enterprise Server, then both are configured independently in the init system. With some init systems, NCP Secure Enterprise Server is automatically started when NCP Secure Enterprise HA Server is started to ensure that this dependency is fulfilled. In some cases, you must ensure that both software are added to the autostart to correctly handle this dependency.

Part of the integration of NCP products into the Linux init system is a configuration file that allows to easily configure parameters for boot up. The following table shows the location of this configuration file for the different init systems:

Table 4: Init system configuration files

Init System	Configuration Location
SystemV	`/etc/default/ncp-<prod>`
Upstart	`/etc/init/ncp-<prod>.override`
systemd	`/etc/sysconfig/ncp-<prod> or /etc/conf.d/ncp-<prod>` (differs between Linux distributions)
OpenRC	`/etc/conf.d/ncp-<prod>`

In the case of the NCP Friendly Net Detection Server, such a configuration file can look like this:

Init Configuration Script for NCP Friendly Net Detection Server on Debian Linux in /etc/default/ncp-fnd

```
# this is an automatically generated init script for NCP Friendly Net
# Detection Server

# You can add command line switches to this variable that shall be passed to
# the sentinel program
SENTINEL_OPTS=""

# You can add command line switches to this variable that shall be passed to
# the control program
CONTROL_OPTS=""

# Allows to pass custom arguments to the ncpfndd daemon process
ncp_args_ncpfndd=""
```

The `ncp_args_*` variables are for passing extra parameters to individual *daemons* as explained here. The `SENTINEL_OPTS` and `CONTROL_OPTS` variables allow to pass custom parameters to invocations of the *sentinel* and *control* programs when executed via the init system.

You can find more advanced information about `<prod>-initconfig` here[19].

5. Command Line Tools

Here you will find documentation about command line tools that are installed together with NCP products under Linux:

Table 5: Overview of command line utilities

Program	Description
`<prod>-config`	A configuration utility installed only for some NCP products that allows interactive and non-interactive configuration of certain software settings. See product specific configuration 22 for more information.
`<prod>-control`	Allows you to control a running instance of an NCP product.
`<prod>-crash`	A tool used to generate and manage information about crashes of NCP programs.
`<prod>-desktopconfig`	A tool only present in NCP Secure Client to perform desktop integration of the graphical program components.
`<prod>-initconfig`	Management of integration in and settings for the Linux init system.
`<prod>-log`	A development tool for getting runtime debugging information from NCP software.
`<prod>-license`	Display and management of active software licenses
`<prod>-sentinel`	Manager process for all *daemon* processes from an NCP product.
`<prod>-uninstall`	Uninstall program

5.1 The sentinel and control Programs

The `sentinel` program is the main program responsible for starting and stopping all the background processes (*daemons*) that belong to an NCP software suite. Some NCP products like NCP Friendly Net Detection Server consist only of a single *daemon* process. The majority, however, consists of a group of several *daemon* processes. Even if NCP software is started via the Linux init system the *sentinel* program is used.

The `<prod>-control` program is the counterpart of the *sentinel* program and can be used to interact with a *sentinel* process running in the background. For example you get information about the current status of the NCP product by calling `<prod>-control -s`. The output shows information about each currently running *daemon* process, such as when it started, what parameters were used to start it, and its process ID.

Checking the status of NCP Friendly Net Detection Server using fnd-control

```
$ fnd-control -s
Current operational status of NCP Friendly Net Detection Server

Friendly Net Detection Daemon
=============================

Status: running since Mo 12 Mai 2014 04:07:16 CDT
Command Line: /usr/local/ncp/fnd/sbin/ncpfndd -f
Process ID: 5010
```

The `<prod>-control` program allows you to shutdown or restart all *daemon* processes run by the *sentinel*. This is done by passing the `--shutdown` or `--restart` switches respectively.

Note

If you have started an NCP product via the Linux init system or the `<prod>-initconfig` tool, do not shut it down in the same way. This can lead to complications because the init system did not prompt the NCP program to do so and therefore assumes that it crashed.

5.1.1 Configuration of Daemons for Automatic Boot up

In the configuration file `sentinel.conf` you can configure the *daemons* that the `sentinel` automatically will start:

Configuration of the sentinel (sentinel.conf)

```
daemons :

{

sem-nginx = false;

};
```

Currently, this file is only used with the NCP Secure Enterprise Management Server. It serves to switch on the web server, which is deactivated by default, for the roll out of the TOTP access data.

The preferred way to achieve this are the command line options `--enable` and --disable of `<prod>-sentinel`.

Activate the web server of the NCP Secure Enterprise Management Server

```
# sem-sentinel --enable sem-nginx
```

If a service is deactivated in the configuration, the command to list all *daemons* will print a note. This command is described in the next section.

5.1.2 Operating on Individual Daemon Processes

You can perform actions on each individual *daemon* process that the sentinel runs. To get a list of all *daemon* processes sentinel knows, call `<prod>-sentinel -l`.

The list of daemon processes for NCP Friendly Net Detection Server

```
$ fnd-sentinel -l
Friendly Net Detection Daemon
=============================

Program call: ncpfndd -f
Description: The single friendly net detection daemon
```

To perform an operation on a *daemon* process you need to identify it by its basename. In the case of NCP Friendly Net Detection Server it is *ncpfndd.* It is the only *daemon* available in NCP Friendly Net Detection Server. The operations you can perform on *daemons* using the `<prod>-control` tool are the following:

- Restarting the specified *daemon*: `--restart-daemon <basename>`
- Disabling the specified *daemon*: `--disable <basename>`
- Enable a previously disabled *daemon*: `--enable <basename>`
- Checking whether the given *daemon* is currently running: `--runs <basename>`

By default the `control` program waits until the requested operation is completed before returning to the command line. Add the `--nowait` parameter to have it returning immediately without waiting for the result of the operation. If you want to set an upper limit on the time waited for a *daemon* to return, specify the `--timeout <seconds>` parameter (60 seconds by default). If the given time is exceeded the respective *daemon* process will be forcibly shutted down. You can also configure the *sentinel* program to exclude certain *daemon* processes from starting in the first place by passing `-x<basename>` or `-o <basename>` to the *sentinel*. `-x` excludes the specified daemon from starting, while `-o` start only the specified *daemon*.

Hinweis

The operations on individual *daemon* processes are only necessary for advanced use or troubleshooting.

Some *daemon* processes are started in multiple different configurations at the same time. This is currently the case for *ncprsud* in NCP Secure Enterprise Management Server. In this case, specifying the `<base name>` is not sufficient to identify a particular instance of a *daemon*. A personality is added to the *daemon* instances in this case. You can determine the different personalities by inspecting the list printed by `<prod>-sentinel -l`. The identification on the command line is then `<basename>:<personality>`. For example specify `ncprsud:radius` to select the radius personality of the *ncprsud daemon* process in NCP Secure Enterprise Management Server.

5.1.3 Controlling How sentinel Deals with Crashes

By default, if any of the *daemons* started by the *sentinel* program exit unexpectedly (e.g. because they crashed), the *sentinel* will shut down any remaining *daemons* and exit. This prevents an incomplete set of services and ensures a clean operating condition.

You can influence in more detail what *sentinel* does in such cases by passing parameters to it. The `--max-crashes` switch defines how many crashes in total the *sentinel* tolerates before shutting down. If you pass *--max-crashes 5* then if more than five crashes occur (any *daemons* that misbehaved are counting too) the *sentinel* will shutdown all processes. Otherwise the crashed *daemon* will be restarted.

In case a *daemon* constantly causes errors (for example, because a configuration file is corrupt), the switches `--max-crashes-per-time` and `--crash-timebase` can be used additionally. These switches allow you to configure a maximum number of crashes within a time period. `--max-crashes-per-time` determines the maximum number of crashes and `--crash-timebase` determines the time period in minutes.

If you want to influence what happens in case of errors, configure a custom script that decides how to handle the situation. For this you pass the `--script <program path>` parameter, where `<program path>` is the path to the executable script that you want to be called in case of a *daemon* misbehaving. The script will be passed a set of environment variables that describe the current situation.

Table 6: Crash script environment variables

Variable	Description	Example Value
`ncp_service`	The NCP *daemon* that crashed	`ncpfndd`
`ncp_crash_code`	The exit code of the crashed *daemon*	`1`
`ncp_crash_signal_nr`	If the *daemon* exited because it received a signal, its number will be provided in this variable	`9`
`ncp_crash_signal_name`	Like `ncp_crash_signal_nr`, but this contains a human readable string naming the signal	`SIGKILL`
`ncp_exit_restart`	The exit code the script should return to have the crashed *daemon* restarted	`N/A`
`ncp_exit_restart_product`	The exit code the script should return to have the complete product restarted in an orderly fashion	`N/A`
`ncp_exit_shutdown`	The exit code the script should return to have the *sentinel* to shut down all remaining processes and exit	`N/A`
`ncp_exit_disable`	The exit code the script should return to have the *sentinel* disable the crashed	`N/A`

	process while keeping the remaining processes running unchanged. This leaves the product in an erroneous state.	
`ncp_exit_internal`	The exit code the script should return to have the *sentinel* fall back to the internal crash logic according to the `--max-crashes`, `--max-crashes-per-time` and `--crash-timebase` switches.	`N/A`

This way it is possible to send an e-mail to inform you about the occurred error or to reboot the Linux system. Note, however, that the *sentinel* program cannot perform additional operations until the crash handling script returns. Here is an example of a *bash* script that could be used for `--script <program path>`:

```
#!/bin/bash

if [ $ncp_crash_code -ne 0 ]; then

        # generally send out an e-mail if a process exited with an error code
        sendmail emergency@mycompany.com <<<"$ncp_service crashed with $ncp_crash_c

fi

if [ $ncp_service = "ncpfndd" ]; then
        # ncpfndd crashed

        if [ $ncp_crash_code -eq 0 ]; then
                # if ncpfndd exited successfully, simply restart it
                exit $ncp_exit_restart

        fi

        # otherwise shut down FND
        exit $ncp_exit_shutdown

fi
```

Upon shutdown of the *sentinel* itself, it will perform a number of cleanup steps to make sure no global state is left behind from crashed *daemon* processes. This could be shared memory areas used for inter-process communication. If such data was left behind, starting the NCP product the next time might fail, because unexpected shared data is found. This is prevented by the *sentinel* performing its cleanup steps. You can also explicitly cause *sentinel* to perform a cleanup by passing the parameter `--clean`. Then the *sentinel* will look for unused global state, remove it and exit without further action.

5.1.4 Passing Custom Parameters to Daemon

You can have *sentinel* pass extra parameters to the individual *daemon* processes it starts. This can be useful for debugging purposes or similar exceptional situations. For example most NCP *daemon* processes support a `--verbose` option to increase their output verbosity. The way to pass such extra parameters is to set environment variables of the pattern `ncp_args_<basename>`, where `<basename>` is the basename of

the *daemon* that should receive the extra parameters. Here is an example for NCP Friendly Net Detection Server:

```
$ export ncp_args_ncpfndd="--verbose"
$ fnd-sentinel -f
```

In this case *ncpfndd* will have the `--verbose` parameter added to its command line when started through *fnd-sentinel*. Add multiple parameters by separating them with spaces in the environment variable.

The configuration file for the init system already contains an entry for passing additional parameters to *ncpfndd*. Likewise, in other NCP products, an environment variable is predefined for each *daemon* process to pass additional parameters to it. Therefore, you only need to add the parameters at the appropriate place when the NCP software is started via the init system.

5.1.5 Accessing Daemon logfiles

Each *daemon* that is started by the *sentinel* is assigned its own logfile in `/var/log/ncp/<prod>/<daemon>.log`. For *ncpfndd* a log file is created in `/var/log/ncp/fnd/ncpfndd.log` for example. The log output is appended to that file, so if the *daemon* is restarted the log file is not overwritten. The sentinel itself logs into `/var/log/ncp/<prod>/sentinel.log`.

If you start the *sentinel* in the foreground by passing the `-f` option the logfiles are not created but the output for all *daemon* processes is written to the console.

5.2 Boot up Configuration using the initconfig Programm

In Boot up of an NCP product using the Linux init system you have already learned the basic use of the `<prod>-initconfig` program. In this section we will look at other features provided by this tool.

The `<prod>-initconfig` program is a tool to handle the different Linux init systems without having to know them in detail. It allows to:

- determine the current configuration of the NCP product regarding the init system
- query the current running status of the NCP product
- start/stop the NCP product via the init system
- add or remove the integration of the NCP product into the init system

5.2.1 Inspecting the Current Configuration

By calling `<prod>-initconfig -i` you will get gathered information about what the configuration status of the NCP product regarding the init system is. This includes:

- whether the product is integrated into the init system
- whether automatic booting is enabled

• whether the product is currently running

If you want to know which files have been installed in the init system for your NCP product, pass `--show-files`. This is also helpful to determine the location of the init configuration file.

Here is an example output for NCP Friendly Net Detection Server:

Information about the init system configuration for NCP Friendly Net Detection Server on Debian Linux

```
$ fnd-initconfig -i
Default Runlevel: 2
NCP Friendly Net Detection Server is currently integrated into UNIX System V
Automatic start on boot is enabled
The product is currently running

$ fnd-initconfig --show-files
/etc/default/ncp-fnd
/etc/init.d/ncp-fnd
```

You can also programmatically query whether the product is integrated or running by passing the `--configured` or `--running` switches and checking the exit code from `<prod>-initconfig`.

5.2.2 Interacting with the Init System

Instead of calling the init system directly by using the commands indicated by the output from `<prod>-initconfig --show-start-command` and `<prod>-initconfig --show-stop-command`, you can use the `<prod>-initconfig` to achieve the same. To start the NCP software via the init system, issue the `--start` and to stop it the `--stop` parameter.

How to change the auto settings is explained here 10.

Finally you can remove the integration of the NCP software into the init system completely via the `--remove` switch and integrate it again using the `--integrate` switch. However, these commands are only used in exceptional cases, such as to restore the original init scripts and init configuration files that were created during the installation of the NCP product.

5.3 Dealing with Software Crashes: the crash Programm

In the event of a program crash, it is important to obtain all available information for NCP support. Only then can our software developers quickly provide a solution to the problem.

For this purpose the `<prod>-crash` program is delivered with each NCP product. On the one hand, it registers itself in the Linux system so that it is called in the event of program crashes and collects all necessary information if an NCP process has crashed. Secondly, it makes it easier for the end user to collect crash information and send it to NCP.

To get an overview of crashes that have occured for a given NCP product, call `<prod>-crash -i`. If you have at least one crash the output will be like the following:

Example list of program crashes for NCP Friendly Net Detection Server

```
$ fnd-crash -i
List of recorded NCP program crashes for NCP Friendly Net Detection Server

Crash of process ncpfndd
========================
Location: /var/adm/ncp/fnd/crashes/ncpfndd.0
Date: Mon 12 May 2014 04:17:40 PM CEST
```

In this case there is a crash for a *daemon* process of NCP Friendly Net Detection Server. The base directory for crash information is `/var/adm/ncp/<prod>/crashes`. For each crash a separate directory is created where crash information from the Linux operating system and additional NCP log files are collected.

You can let the `<prod>-crash` program create a compressed archive containing all currently known information about crashes for the respective product. This is done by passing the `--report` switch and the path where the archive should be written to.

Generating a crash report archive for NCP Friendly Net Detection Server

```
$ fnd-crash --report /tmp
The crash report file has been successfully created at
'/tmp/fnd_crash_report1.tar.bz2'.
```

You can send the resulting file to the NCP support if an error situation occurs.

Crash reports can contain sensitive information such as user names, e-mail addresses or even parts of secret key material. For your security, these crash reports will be stored encrypted in the current version of NCP products, so that only authorized NCP employees can view this data.

If you want to disable this encryption, then pass the parameter `--no-encryption`.

The program `<prod>-crash` also allows the output of data about the running system. These can be useful information for the NCP support. You can obtain this information via the command line `<prod>-crash --system-info`.

5.3.1 Deleting Old Crash Reports

To limit the disk space that is occupied by crashdumps, saving new dumps automatically deletes older crash information. It is possible to initiate the deletion of dumps by calling `<product>-crash --delete-old` manually. All reports that exceed the number `max_count` or are older than `max_age` days are deleted.

This parameters can be adjusted in the configuration `file global.conf` of the appropriate product. If both values are 0, no crashdumps are deleted.

Configuration of the maximum kept crash records (global.conf)

```
crashdump:
{
     max_count = 20;
     max_age = 30;
};
```

5.4 Dealing with Product License and Version Using the license Programm

Most NCP products require a purchased license key for full functionatily. An exception to this is NCP Friendly Net Detection Server. All products come with a 30 day trial period. After that period you are required to register a valid license for the product to function.

To inspect the current license data a separate utility called `<prod>-license` is provided. The tool displays the remaining time the license is valid and some additional information depending on the active license and product. Here is an example for NCP Secure Enterprise Server using a trial license that is valid for five more days:

Inspecting the active license for NCP Secure Enterprise Server

```
$ ses-license
   >>>>> Current license data <<<<<
   Software version: NCP Secure Enterprise Server 8.14 (experimental)
   Licensed version: trial version
   Valid for: 5 days
```

In all products except NCP Secure Enterprise Management Server, a full license is not activated via the command line, but via the web interface (NCP Secure Enterprise Server and NCP Secure Enterprise HA Server) or via the monitor application (NCP Secure Client).

For NCP Secure Enterprise Management Server, however, a license is activated or updated using the `sem-license` utility. You can either call it with the `--activate` parameter, which will cause the program to interactively query the license data to activate. Alternatively you may pass the license data via the `--license` parameter that takes the license in form of `<key>:<serial>`, where `<key>` is a 5 x 4 digit key separated by minuses and the `<serial>` is an 8 digit serial number.

Newer versions NCP Secure Client also allow the input of a license key this way as an alternative to the graphical user interface.

6. Product Specific Configuration

This section covers utilities and configuration tasks that are specific to individual NCP products.

6.1 NCP Secure Client

6.1.1 Adding Desktop Icons and Menu Entries Using clnt-desktopconfig

The `clnt-desktopconfig` utility performs integration of NCP Secure Client into the graphical desktop. Depending on the desktop you are using, this includes creation of desktop icons and menu entries to start the graphical monitor application.

Every user in the system, who wants to use the NCP Secure Client, can call `clnt-desktopconfig`. The prerequisite is that the user is a member of the installation group of NCP Secure Client (*ncp* by default).

The program performs the desktop integration for the calling user. This means you cannot generate desktop icons as *root* for a different user.

The basic program switches `clnt-desktopconfig` supports are `--remove` and `--integrate`, which remove or integrate the NCP Secure Client into the callers desktop environment, respectively.

Note

You can only run the graphical monitor application when the NCP Secure Client *daemons* are running in the background.

6.2 NCP Secure Enterprise Server

6.2.1 SNMP Setup

You can use the SNMP protocol (Simple Network Management Protocol) for retrieving and processing information about the operational state of NCP Secure Enterprise Server over the network. This can be used for monitoring.

Under Linux, this requires the installation of the service `snmpd` (the term `net-snmp` is also used), which implements the SNMP protocol. You can install it via the package management of your Linux distribution. For the general setup and usage of `snmpd` refer to the information delivered with your Linux distribution or provided on the project websites on the Internet.

For the `snmpd` server being able to receive data from NCP Secure Enterprise Server it is necessary to adjust a configuration entry. Typically the relevant configuration file for `snmpd` is located in `/etc/snmp/snmpd.conf`. There you add the following line:

Entry of the NCP Secure Enterprise Server SNMP plugin into snmpd.conf

```
dlmod ncpSecureServer /opt/ncp/ses/lib/libncpsrvagent.so
```

Please note that you must adjust this path if you have installed NCP Secure Enterprise Server in a different directory.

After the complete configuration of `snmpd` you have to start or restart the service. If the configuration and access rights have been set up correctly, the following command should provide a list of status values of the NCP Secure Enterprise Server:

Test query of NCP SNMP data after the finished setup of snmpd

```
snmpwalk -v 1 -c public localhost iso.3.6.1.4.1.1213.8
```

6.3 NCP Secure Enterprise HA Server

6.3.1 SNMP Setup

The setup of SNMP for NCP Secure Enterprise HA Server is done analogous to the explanation found in SNMP for NCP Secure Enterprise Server. You only need to use the following line for setting up the plugin:

Entry of the NCP Secure Enterprise HA Server SNMP plugin into snmpd.conf

```
dlmod ncpHaSrv /opt/ncp/has/lib/libncphasrvagent.so
```

6.4 NCP Secure Enterprise Management Server

NCP Secure Enterprise Management Server requires more local configuration than the other products, as the server requires a database connection before it can start. This means that it must be configured locally and not through the management console.

The basic settings of NCP Secure Enterprise Management Server can be configured using the `sem-config` program. The program requires *root* privileges, if necessary the user will be asked for the *root* password or their own password at startup. We deliberately chose to use a text interface instead of a full GUI, as this also works via an SSH connection or directly via the Linux console and can therefore also be used on servers without a desktop manager.

Updating from older versions

The text-based configuration tool `sem-config` described here was introduced in version 5.30 of NCP Secure Enterprise Management Server. The program was available before this version but it could only be used to change the operation mode, either interactively or in batch mode. All other settings had to be configured manually in the configuration file.

For older versions, please refer to the documentation distributed with the respective version. The description of the configuration file options has been removed in this version of the documentation. The file can still be edited manually. The batch mode of the tool for switching the operation mode works as previously without the need to adapt existing scripts.

The program is based on the management configuration under Windows. The user interface and the terminology used in the Linux version are based on this application. When the program is started, the main menu is displayed. This menu corresponds to the tabs in the Windows application.

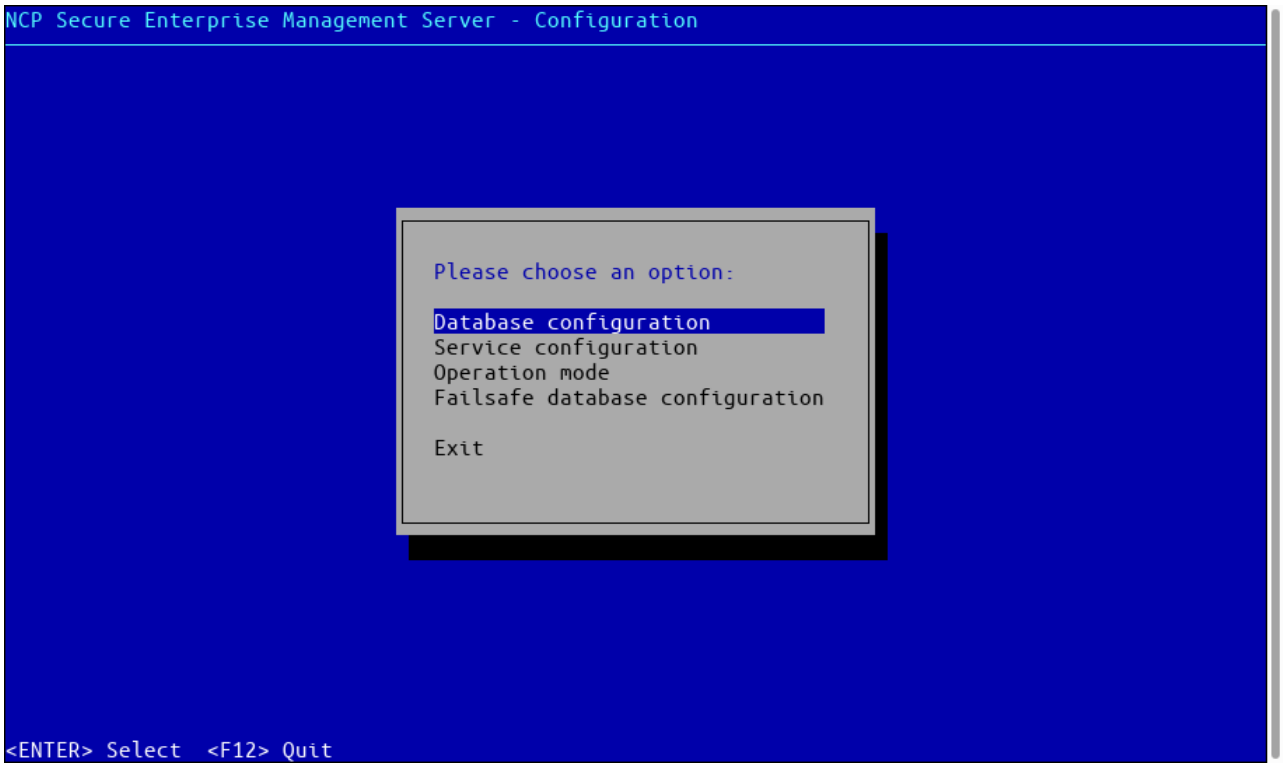

Figure 1: Main menu

Use the cursor keys to select a menu item and confirm your selection with the Enter key. To end the program, you can either select Exit and confirm with the Enter key or press the F12 key.

The cursor keys can be used for navigation. The Tab key jumps to the next input field, radio button or button, the space bar selects an option, the Enter key confirms. The function keys are displayed at the bottom of the screen. All functions are additionally available via buttons if the function keys do not work as expected (for example via SSH).

6.4.1 Database Configuration

The program allows you to configure both the primary database and the database for the failsafe server. The settings are identical, therefore only the primary database is shown here; they also apply to failsafe servers.

Two interfaces are supported to access the database:

1. The native connection for MariaDB- and MySQL databases
2. The ODBC interface via the unixODBC compatibility layer

However, before the database can be configured and tested using `sem-config`, a database must first be created. This is described in the next section. If the database already exists, you can skip this section.

Database setup

Regardless of whether the access was configured via the native connector or via ODBC, the database itself still has to be set up. The complete setup of the MySQL database goes beyond the scope of this documentation. Please refer to the documentation of your Linux distribution for the MySQL server for more information on this topic.

Note

User accounts with empty passwords are not supported by NCP Secure Enterprise Management Server.

Once the MySQL server is set up and running correctly, you must now log into the MySQL server console and create the database. This can be achieved as follows:

Create an empty database named semdb

```
$ mysql -u root -p
Enter password: <password>
Welcome to the MySQL monitor. Commands end with ; or \g.
[...]

mysql> create database semdb;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
```

This variant uses the MariaDB Connector/C or MySQL Connector/C (both are interface compatible) to communicate with the database.

This is the recommended variant if a MariaDB or MySQL database is used. The unixODBC interface should only be used for other database platforms.

The configuration interface is shown in Figure 2.

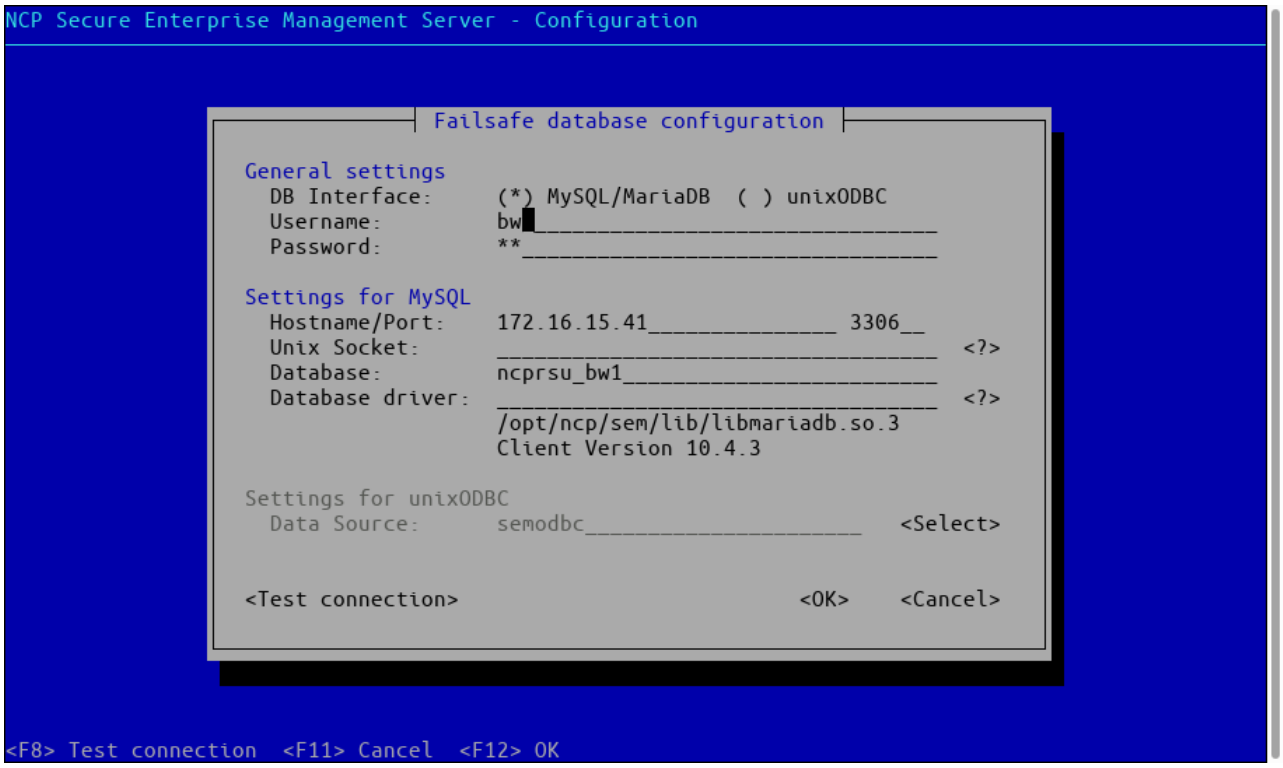

Figure 2: Configuring a MariaDB database

If not already selected, select the MariaDB interface by selecting the (*) MySQL/MariaDB radio button with the Tab key and activate it with the space bar.

The Unix socket is an alternative to TCP/IP if the database and the management server are on the same computer. The path to a Unix domain socket is entered in this field. This is specific to the Linux distribution. If the `mysql` or `mariadb` command line tools are set up, they can be used to find the socket path:

Finding the unix domain socket path with MySQL

```
$ mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
...
mysql> show variables like 'socket';
+---------------+-----------------------------+
| Variable_name | Value                       |
+---------------+-----------------------------+
| socket | /var/lib/mysql/mysql.sock          |
+---------------+-----------------------------+
1 row in set (0.00 sec)
mysql> Bye
```

To use the Unix domain socket, the hostname must be configured to `localhost`. If this field is empty, local TCP/IP communication is used.

The database driver is a library (Shared Object) that implements the communication with the MariaDB or MySQL database. The database driver is Connector/C. NCP has included the latest version of the MariaDB Connector/C in each release since Version 5.30. If nothing is entered in this field, this driver is used. It is still possible to configure a driver that has been installed manually or with the Linux distribution.

Enter the absolute path or the file name only (this searches in the system library directories) for the database driver in this field. When you leave this field, the system immediately checks whether the file is a valid database driver and displays the version. There is no communication with the database yet.

To test the settings, use the Tab key to select the *Test connection* button or press F8. If the connection is successful the message displayed in Figure 3 will be shown. In the event of an error, the message displays more detailed information about the cause of the error.

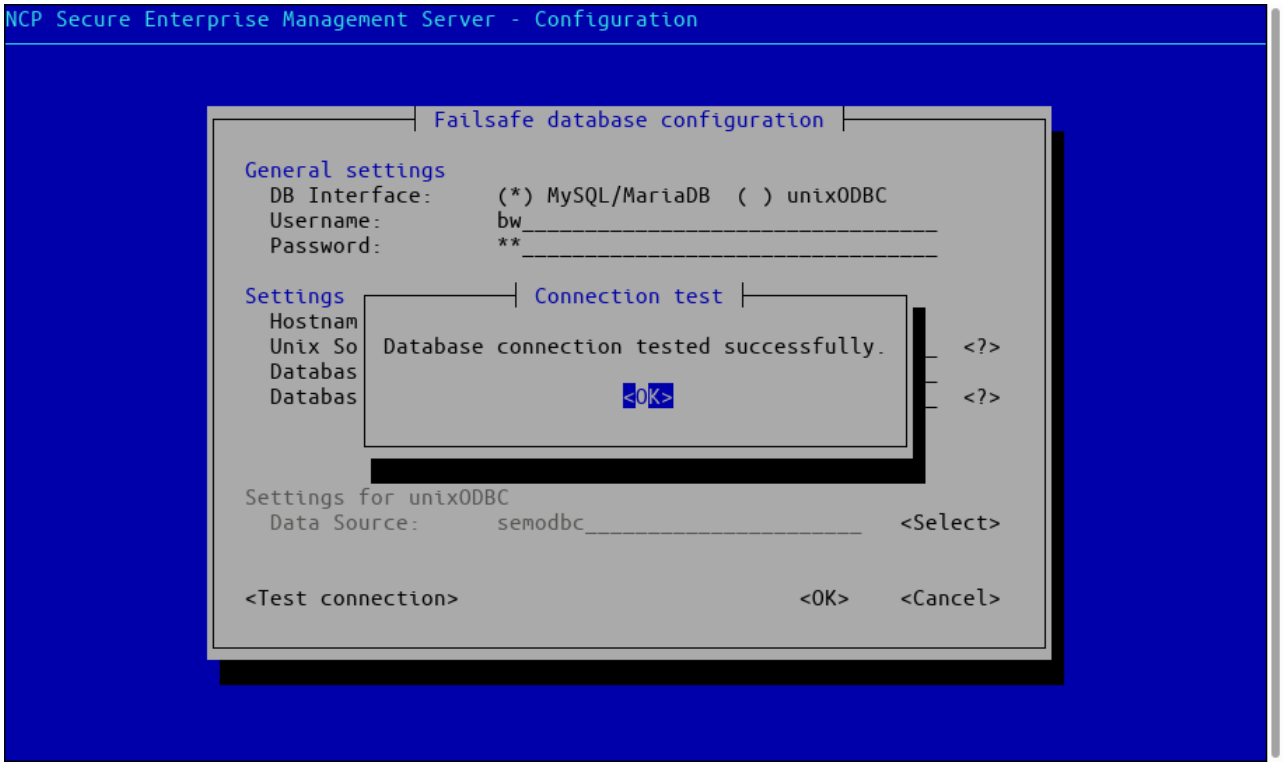

Figure 3: Successful connection test

The configuration is not yet saved for testing the connection. This only happens if you leave the dialog by confirming with *OK* or pressing F12. Press *Cancel* or F11 to exit without saving.

Database configuration using unixODBC

In this case, the *ODBC* interface (Open Database Connectivity) is used for connecting to the database. This is a software layer that mediates between the actual database and NCP Secure Enterprise Management Server.

To configure the database and the ODBC interface, you need to install and prepare some packages on your Linux distribution. Although different database connections and ODBC interfaces are supported in principle, the most common configuration is as follows:

- MySQL for the database
- the library unixODBC as ODBC interface
- the driver myodbc to link unixODBC with MySQL

In this guide, we assume that you have not yet set up any of these packages. Typically, you can install this software via the package manager of your Linux distribution. The following table gives an overview of the package names and commands to install the packages on common Linux distributions:

Table 7: MySQL / ODBC package installation under Linux

Linux distribution	Package names	Installation command
Debian, Ubuntu	`mysql-server, unixodbc, libmyodbc`	`apt-get install mysql-server unixodbc libmyodbc`
Red Hat, CentOS	`mysql-server, unixodbc, mysql-connector-odbc`	`yum install mysql-server unixodbc mysql-connector-odbc`
SUSE SLES	`unixODBC, mysql, MyODBC-unixODBC`	`zypper install unixODBC mysql MyODBC-unixODBC`

Note

It may be necessary to configure additional repository sources in SUSE to obtain the MyODBC package.

If the *mysql* database has been replaced by the alternative *mariadb* database in your Linux distribution, most of the instructions given here will apply. In this case you only have to install the *mariadb* packages instead of the *mysql* packages. The *unixodb* and *myodbc* packages remain the same.

Depending on the Linux distribution, new configuration files will be located either in `/etc` or in `/etc/unixODBC` after the successful installation of all necessary packages. The files are named `odbc.ini` and `odbcinst.ini`. You will need to customize these files to suit your environment.

The ODBC driver is configured in `odbcinst.ini`. A name for the driver configuration must be given and the path to the driver library must be specified. In the following example we use the name `myodbc`. The path to the driver library may differ between Linux distributions:

Example of odbc.ini

```
[semodbc]
Driver = myodbc                                    ①
Description = MySQL connection for NCP-SEM
Server = localhost                                 ②
Port = 3306
Database = semdb                                   ③
```

① This is the name of the driver configuration as specified in odbcinst.ini

② This is the default server and port for the locally running MySQL server

③ This is the name of the database within MySQL that will be used by NCP Secure Enterprise Management Server

Now start `sem-config` and select Database configuration. Change the connection type to *unixODBC* by selecting the corresponding
selection field with the Tab key and activating it with the space bar.

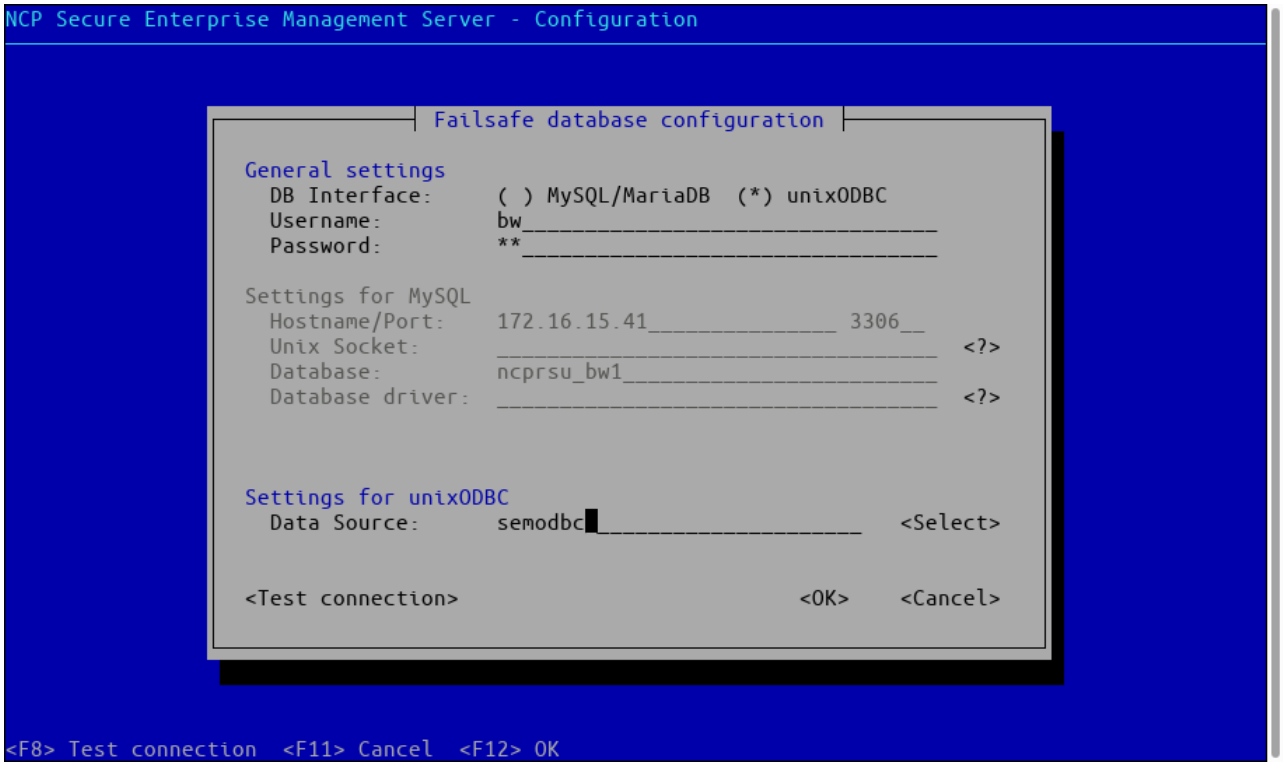

Figure 4: Configuring the database with unixODBC

Most of the fields are grayed out as they have already been defined in the ODBC configuration. You only have to enter the username, the password and the data source from `odbc.ini`. Use the *Select* button to list all data sources and select the one you need.

To test the settings, use the Tab key to select the *Test connection* button or press F8. If the connection is successful the message displayed in Figure 3 will be shown. In the event of an error, the message displays more detailed information about the cause of the error. The configuration is not yet saved for testing the connection. This only happens if you leave the dialog by confirming with *OK* or pressing F12. Press *Cancel* or F11 to exit without saving.

Testing the connection via the command line

```
$ ncprsud -testDB
Init Database Connection
Database Connection ok
Begin Test Database Access and Types
[...]
```

Use the connection test to display any error messages completely and redirect them to a file if necessary.

6.4.2 Service Configuration

NCP Secure Enterprise Management Server consists of several services that interact with each other. The services required for operation depends essentially on the operation mode 32. The services are started and stopped by the sentinel program 14, so they are only indirectly known to the Init system.

The only service that should normally be enabled or disabled manually is the web server *sem-nginx*, which provides the web page that users access directly to retrieve their TOTP credentials. If this feature is not required, the web server does not need to be running. Since this applies to the majority of installations, it is also deactivated in the default configuration.

The services configuration is shown in Figure 5.

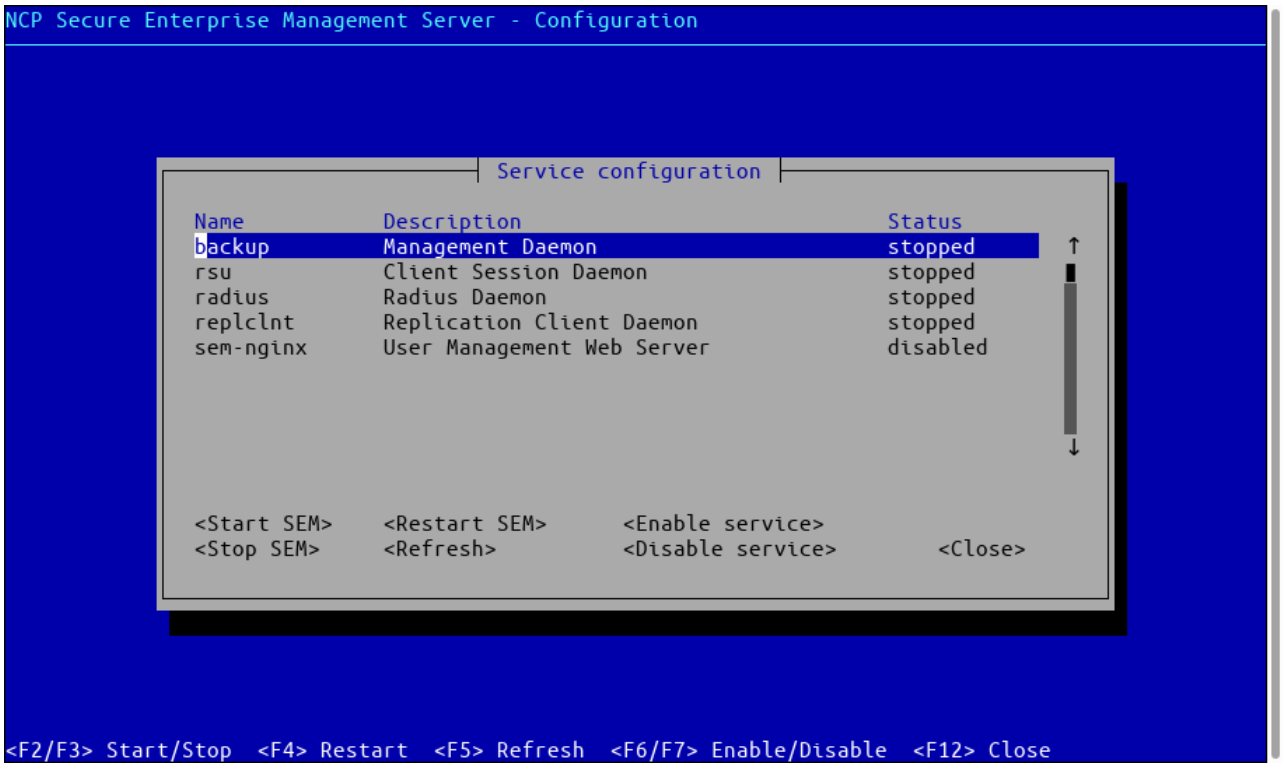

Figure 5: Configuration services

The status of the service is displayed in the upper part of the dialog. Besides *stopped* and *running* there is also the additional state *disabled*, which means that it does not run and is not started when the master *daemon sentinel* is started. The following functions are available via the corresponding selection fields or function keys:

- Start SEM (F2): The NCP Secure Enterprise Management Server is started via the Init system if it is not already running. This function is equivalent to `sem-initconfig --start`.
- Stop SEM (F3): If Management-Server is running, it is ended by the Init system. This function is equivalent to `sem-initconfig -- stop`.
- Restart SEM (F4): Management-Server will be restarted if it is running. The master daemon Sentinel is not restarted. This function is equivalent to `sem-control --reload --restart`.

- Update (F5): The status of the services in the display is updated. This is particularly necessary if the state has changed without the change being caused by this program.
- Enable (F6): Immediately enables and starts a service which has been disabled. This function is equivalent to `sem-sentinel --enable <service>` (activates the service) followed by `sem-control --enable <service>` (starts the service immediately and not at the next start).
- Disable (F7): Immediately disables a service which has been disabled. This function is equivalent to `sem-sentinel --disable <service>` (deactivates the service) followed by `sem-control --disable <service>` (ends the service immediately and not at the next start).

6.4.3 Configuring the Operation Mode

As described in the product documentation, Management-Server has three operating modes:

- *Primary Server:* The NCP Secure Enterprise Management Server in primary mode manages the master database.
- *Backup Server:* Can be used as a read-only mirror of the primary server.
- *Failsafe Server:* A server running in backup mode can be put into this mode to recover a failed NCP Secure Enterprise Management Server in primary mode. It then assumes the role of the primary server until the original server is available again.

`sem-config` enables both the switching of the operation mode and the initial configuration. Figure 6 shows the configuration in primary mode, Figure 7 shows the configuration in backup mode. The changes are only applied after a restart. If Management-Server is running when you confirm with OK, you will prompted automatically to restart the service.

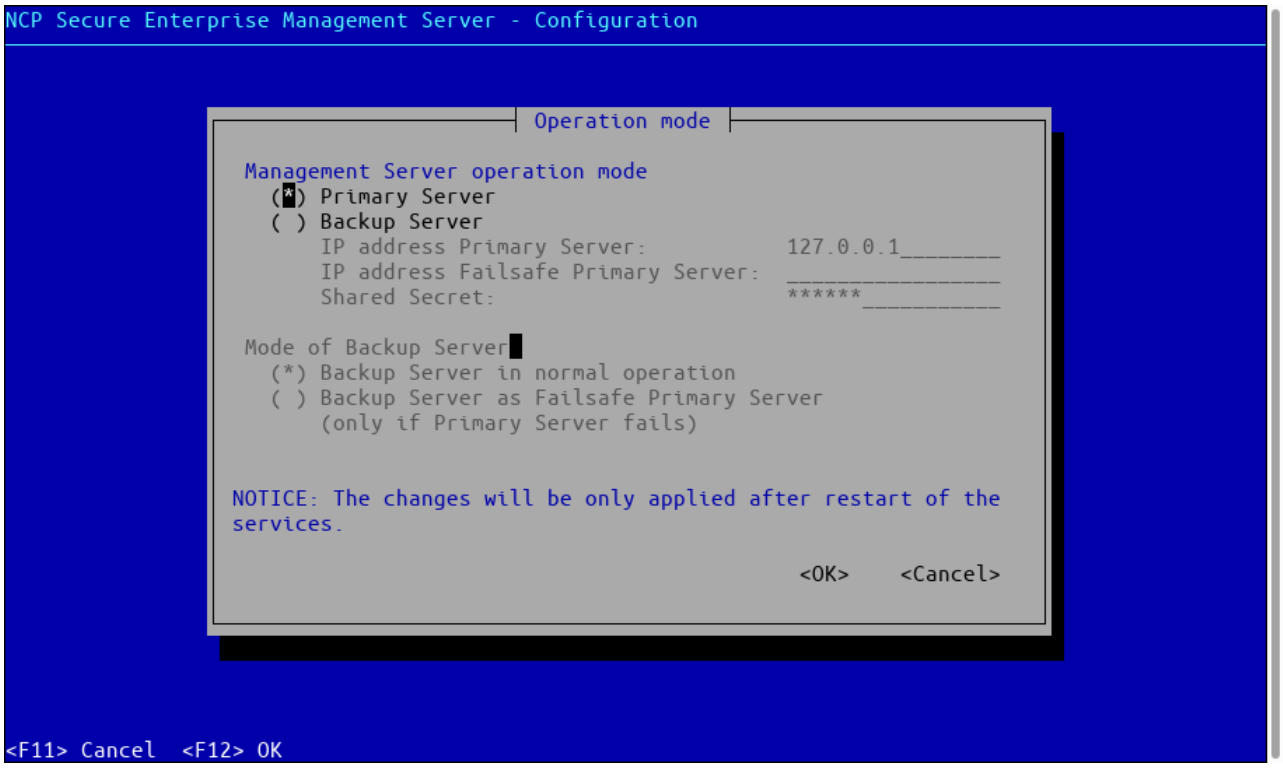

Figure 6: Configuring services in primary mode

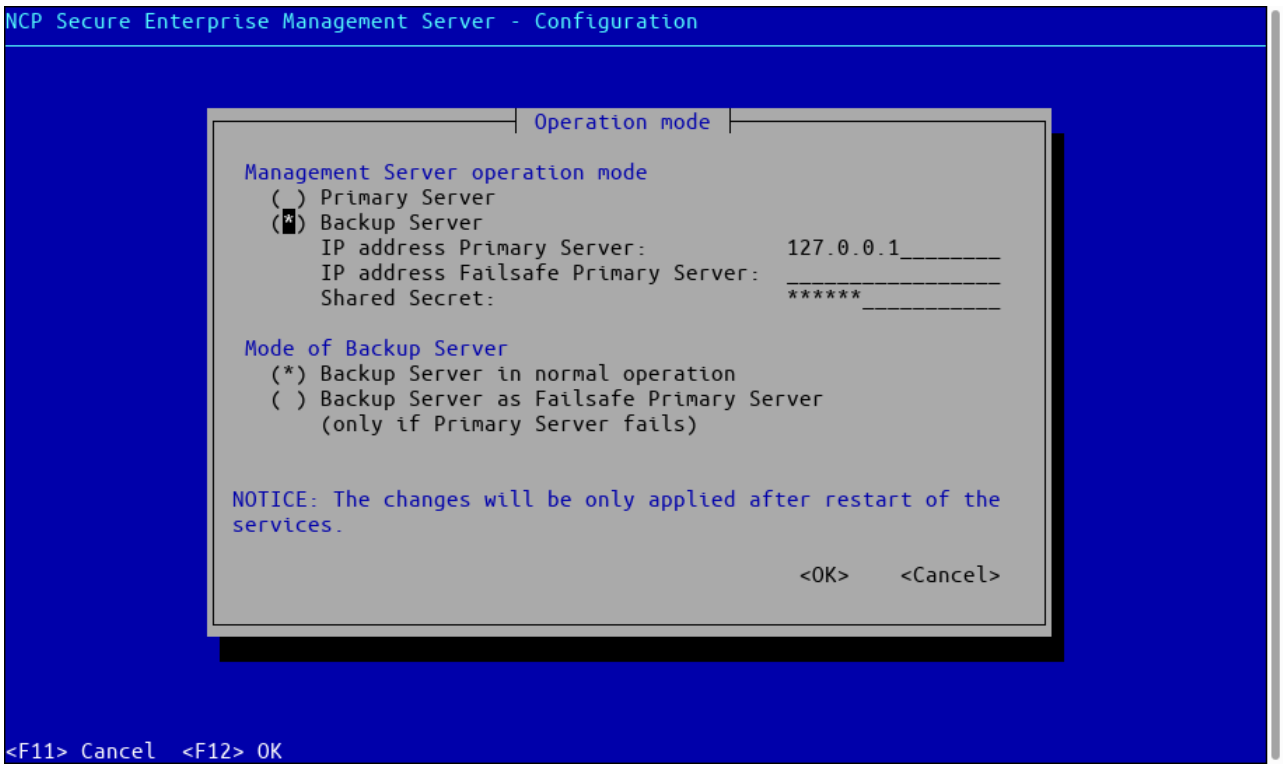

Figure 7: Configuring services in backup mode

Switching between backup mode and failsafe mode in batch mode

You can switch between the Backup and Failsafe operating modes using the switch `--mode <BACKUP|FAILSAFE>` without having to start the text interface

```
$ sem-config --mode=FAILSAFE
```

The changes only take effect after a restart, which must be done manually, either via the Init system or with `sem-control --reload --restart`. The `--reload` switch causes the *sentinel* service to reload its configuration – and thus the SEM mode – before restarting the programs. If the operation mode is changed, other services may have to be started.