Dr.WEB®

Anti-virus
for Novell Storage Services

Defend what you create

Administrator Manual

Dr.Web® Antivirus for Novell Storage Services
Version 6.0.1
Administrator Manual
15.01.2013

Doctor Web Head Office
2-12A, 3rd str. Yamskogo polya
Moscow, Russia
125124

Web site: www.drweb.com
Phone: +7 (495) 789-45-87

Refer to the official web site for regional and international office information.

Doctor Web

Doctor Web develops and distributes Dr.Web® information security solutions which provide efficient protection from malicious software and spam.

Doctor Web customers can be found among home users from all over the world and in government enterprises, small companies and nationwide corporations.

Dr.Web antivirus solutions are well known since 1992 for continuing excellence in malware detection and compliance with international information security standards. State certificates and awards received by the Dr.Web solutions, as well as the globally widespread use of our products are the best evidence of exceptional trust to the company products.

We thank all our customers for their support and devotion to the Dr.Web products!

Table of Contents

Introduction

Manual is designed for the person responsible for anti-virus protection and security ("Administrator" hereinafter).

Dr.Web® Antivirus for Novell Storage Services serves for detection and neutralization of viruses and other malware in **Novell Storage Services™** (**NSS**) file system based on **Novell Open Enterprise Server™** running under **SUSE Linux Enterprise Server™** 10 SP3 operating system. Though most malware is made for non-UNIX systems, file servers can be used for spreading viruses for all operating systems including macro-viruses for applications.

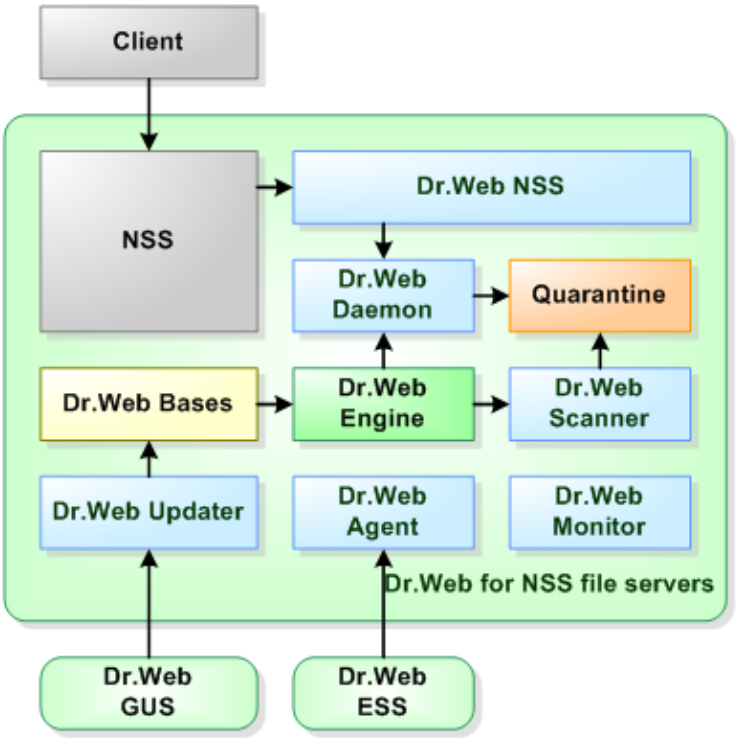
Dr.Web for Novell Storage Services is able to detect all known viruses and works in asynchronous mode: files are processed without locking. Checking for viruses is made when server performs requested file operation (i.e writing or reading files on the server).

The following modules are included into the **Dr.Web for Novell Storage Services** solution:

- **Console antivirus scanner Dr.Web Scanner** used to detect and cure viruses on the local machine and shared directories;
- **Background resident component Dr.Web Daemon** used as an external antivirus filter;
- **Auxiliary resident component Dr.Web Monitor** used to run and terminate other **Dr.Web** modules in the necessary order;
- **Component Dr.Web Agent** used for gathering statistical information and integration with **Dr.Web Enterprise Security Suite**;
- **Component Dr.Web Updater** (it is realized in the form of a Perl script) used to automatically update virus databases;
- **Background component Dr.Web NSS** – main module responsible for integration with NSS file system;.

Structure of **Dr.Web for Novell Storage Services** and its components are shown on figure below.

Figure 1. Structure of Dr.Web for Novell Storage Services and its components



In the present manual basic steps of setup, adjustment and startup procedures of **Dr.Web for Novell Storage Services** solution will be discussed. This manual contains information on the following topics:

- General product description;
- Installation of **Dr.Web for Novell Storage Services** solution;
- Running **Dr.Web for Novell Storage Services** solution;
- Usage of updating package **Dr.Web Updater**;
- Usage of **Dr.Web Agent**;

- Usage of console scanner **Dr.Web Scanner**;
- Usage of background on-demand scanner **Dr.Web Daemon**;
- Usage of **Dr.Web Monitor**;
- Usage of **Dr.Web for Novell Storage Services** file monitor.

At the end of this Manual you will find technical support service contact information.

Doctor Web products are being constantly developed. Add-ons to virus databases are released daily or even several times a day. New versions of programs appear. Diagnostics techniques and methods of anti-virus protection, as well as integration with other applications of UNIX systems are improved regularly. Besides that, the list of applications compatible with **Doctor Web** products is constantly expanding, therefore some settings and functions described in this Manual will slightly differ from current program version. To get up-to-date program information please refer to documentation files included in delivery package.

Terms and abbreviations

This guide utilizes the following content conventions and signs:

Convention	Description
Bold	Names of buttons and other elements of the graphical user interface (GUI), and required user input that must be entered exactly as given in the guide.
Green and bold	Names of **Doctor Web** products and components.
Green and underlined	Hyperlinks to topics and web pages.
`Monospace`	Code examples, input to the command line and application output.
Italics	Placeholders which represent information that must be supplied by the user. For command-line input, it indicates parameter values.
CAPITAL LETTERS	Names of keys and key sequences.

Convention	Description
Plus sign ('+')	Indicates a combination of keys. For example, ALT+F1 means to hold down the ALT key while pressing the F1 key.
Exclamation mark	A warning about potential errors or any other important comment.

To define directories to which components of the software complex are installed, specific conventional symbols are used: `%bin_dir`, `%etc_dir` and `%var_dir`:

```
%bin_dir = /opt/drweb/
%etc_dir = /etc/drweb/
%var_dir = /var/drweb/
```

The following abbreviations are used in this Manual:

Abbreviation	Description
ASCII	American Standard Code for Information Interchange
CIDR	Classless Inter-Domain Routing
DEB	Extension for package files for software distribution in **Debian** (and others used **dpkg**)
DNS	Domain Name System
HTML	HyperText Markup Language
IP	Internet Protocol
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
IPC	Inter-Process Communication
MD5	Message Digest 5 algorithm
OS	Operating System
PID	Process IDentifier in UNIX based OS
POSIX	Portable Operating System Interface for Unix
RFC	Request for Comments

Abbreviation	Description
RPM	Package files format (and extension) for **Red Hat Package Manager**
SSL	Secure Socket Layers protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security protocol
URL	Uniform Resource Locator
UUID	Unique User IDdentifier
XML	eXtensible Markup Language

System requirements

Dr.Web for Novell Storage Services requires

- **Novell Open Enterprise Server SP2** based on **SUSE Linux Enterprise Server 10 SP3** operating system;
- Installed **Novell Storage Services** (**NSS**);
- NSS file system mounted to the specified directory;
- **Dr.Web Updater** component is require installed **Perl** 5.8.0 or later.

Hardware requirements are the same as for **SUSE Linux Enterprise Server 10 SP3** operating system.

Dr.Web for Novell Storage Services installation requires at least 300 megabytes of free space.

Graphic installer and uninstaller of **Dr.Web for Novell Storage Services** require **X Window System**. Interactive configuration script requires terminal emulator **`xterm`** or **`xvt`** for running in graphic mode.

Depending on the scope of problems to be solved by **Dr.Web for Novell Storage Services** and system load, hardware requirements may rise.

Compatibility with Linux Distributions

Dr.Web for Novell Storage Services solution is compatible with **Linux** distribution **SUSE Linux Enterprise Server 10 SP3**.

Package files location

Dr.Web for Novell Storage Services solution is installed by default to `%bin_dir`, `%etc_dir` and `%var_dir` directories. OS-independent directory tree is created in these directories:

- `%bin_dir` – executable modules of **Dr.Web for Novell Storage Services** and updating package **Dr.Web Updater** (perl script `update.pl`);
- `%bin_dir/lib/` – anti-virus engine as loadable library (`drweb32.dll`). In the same subdirectory various service libraries for packages of **Dr.Web for Novell Storage Services** solution can reside;
- `%etc_dir/agent/` – additional configuration files for **Dr.Web Agent** module;
- `%etc_dir/monitor/` – additional configuration files for **Dr.Web Monitor** module;
- `%var_dir/bases/*.vdb` – databases of known viruses;
- `%etc_dir` – configuration files of **Dr.Web for Novell Storage Services** solution: `drweb32.ini`, `agent.conf`, `monitor.conf`, `drwebd.enable` and `drweb-monitor.enable` (two last-mentioned are for adjustment of daemons' operation*);
- `%bin_dir/lib/ru_scanner.dwl` – language file for **Dr.Web Scanner** package;
- `%bin_dir/scripts/` – interactive configuration script, migration script for updating configuration from older versions;
- `%etc_dir/templates/` – templates for notifications to be sent on detection of malicious objects or errors during scanning and file processing;

- `%bin_dir/doc/` — documentation. All documentation is presented in plain text files in English and Russian (KOI8-R and UTF-8 encodings) languages;
- `%var_dir/infected/` – **Quarantine** directory to move infected or suspicious files to, if such reaction is specified in settings for **Dr.Web for Novell Storage Services** software system components.

*) Placement of the `enable` files is depends from **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:

  Files will be placed to directory `%etc_dir` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

- Installation from **native DEB packages**:

  Files will be placed to directory `/etc/defaults` and will be named
  `drwebd`,
  `drweb-monitor`.

- Installation from **native RPM packages**:

  Files will be placed to directory `/etc/sysconfig` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

Configuration files

All **Dr.Web for Novell Storage Services** settings are stored in configuration files which you can use to configure all software components. Configuration files are plain text files in the following format:

```
--- beginning of the file ---

[Section 1 name]
Parameter1 = value1, ..., valueK
...
ParameterM = value1, ..., valueK

[Section X name]
Parameter1 = value1, ..., valueK
...
ParameterY = value1, ..., valueK

--- end of the file ---
```

Configuration file content is satisfy to the next rules:

- Symbol ';' or '#' in line of configuration file mark a text in part of line from this symbol to end of line as comment. This text in line will be ignored by a **Dr.Web for Novell Storage Services** modules while they will read configuration from a file.
- Contents of the file divide on a set of the named sections. Possible names of sections are rigidly determined and can't be changed. The name of section is set in square brackets.
- Each section of the file contain a group of the semantic parameters, which united on sense.
- In one line of file may be specified value (or values) only of one parameter.
- General format of parameter's value setting (spaces around '=' will be ignored):

  ```
  <Parameter name> = <Value>
  ```

- Names of parameters are rigidly determined and can't be changed.
- Names of all sections, parameters, and values in file are case insensitive (only if value is not a directory or file name, because in UNIX systems they are case sensitive).
- Order of sections in file and order of parameters in sections

are has no importance.

- Values of parameters in file may be enclosed in quotation marks (and must be enclosed in quotation marks if it contains white spaces).
- Some parameters can have more than one value. In this case values of parameter are separated by a comma or value of parameter is set several times in different lines of the configuration file. At transfer of values of parameter through a comma spaces between value and a comma (if they are present) are ignored. If the white spaces is a part of the value, all value is necessary for quoting.

Assignment possibility to parameter some values in this document is specified obviously. If for some parameter in this document or in comments in the configuration file obviously it is not specified that it can appropriate some values, the parameter may have only one value.

Example of setting of parameter with more than one value:

1) Comma-separated list of parameter's values:

```
Parameter = Value1, Value2,"Value 3"
```

2) Setting of parameter's values several times in different lines of the configuration file:

```
Parameter = Value2
Parameter = Value1
Parameter = "Value 3"
```

Please note, that order of values assignment to parameter has no importance.

If any parameter is commented out or not specified, it does not mean that this parameter has no value. In this case the default value will be used. Only few parameters are optional or do not have default values. All such cases will be described separately.

Notation of parameters description that is used in this Manual

Description of each parameter in this manual looks like as:

ParameterName = {Parameter type | Possible values}	Description of a parameter {Whether can have more than one value} {Special remarks} {Important remarks} Default value: **ParameterName** = {value | nothing}

Parameters in document are described in the order they are presented in the respective configuration file of **Dr.Web for Novell Storage Services**.

In configuration files of **Dr.Web for Novell Storage Services** are used followed parameter types:

- **numerical value** — parameter value can be zero or natural number.
- **time** — parameter value is time in selected units. Value is combine of integer (non-negative number of time units) and one symbol which is determine type of unit (`s` – seconds, `m` – minutes, `h` – hours, symbol is case insensitive). If symbol is not presented in the value, then type of time unit is seconds (by default).

  Examples: `30h`, `15m`, `6` (in last example – seconds).
- **size** — parameter value is size of memory block (on disk or in RAM) in selected units. Also combine non-negative integer number of units and one symbol which is determine type of unit (`b` – bytes, `k` – kilobytes, `m` – megabytes, `g` – gygabytes, symbol is case insensitive). If symbol is not presented in the value, then type of memory block unit is bytes (by default).

  Examples: `20b`, `15k`

- **permissons** — parameter value is three-digit integer which determine file access permissions in UNIX format:
  Each permission (digit) in these value is a combination (sum) of three base permissions:
  - Read permission (r) - 4;
  - Write permission (w) - 2;
  - Execute permission (x) - 1.

  First digit in value is determine permissions for file owner, second for owner's group, and third for all other users (not owners and not members of owner's group).

  Examples: `755`, `644`

- **logical** (**Yes/No**) — parameter value is string with logical value "`Yes`" or "`No`".

- **path to file/directory** — parameter value is string which determine path to a file or a directory in a file system. Note, that in UNIX systems files and directory names are case sensitive. If it is specified that the **mask** can be value of parameter, then you can to specify the file masks, which can contain the followed special symbols:
  - `?` – replaces any one symbol in a name of the file (directory);
  - `*` – replaces any (including empty) sequence of symbols in a name of the file (directory).

  Example: "`?.e*`" – the mask with which satisfy the files which name consists of only one any symbol, and extension of any length and begins with the symbol '`e`' (`x.exe`, `g.e`, `f.enable` and so on).

- **action** — parameter value is string with name of action (reaction of **Dr.Web for Novell Storage Services**) which is must be applied to some object in dependence of results of its scanning. In some cases parameter can have more than one action (in this case parameter's type named **actions list**). In this case first action is mandatory and next actions are optional. Actions list can contain from 0 to 3 optional actions. Set of allowed actions can be different for different parameters (set of allowed actions is presented in

Manual for each parameter). Common set of allowed actions see in chapter Allowed actions.

- **address** — parameter value is string with socket's address of some component of **Dr.Web for Novell Storage Services** or of used external program (for IPC).
  Address always presented in a format TYPE:ADDRESS. Next TYPEs are allowed:
  - inet — this is a TCP socket, ADDRESS is specified in PORT@HOST_NAME format, where HOST_NAME can be either direct IP-address or host domain name.
    Example:
    ```
    Address = inet:3003@localhost
    ```
  - local — this is a local UNIX socket, ADDRESS in this case – path to a socket file.
    Example:
    ```
    Address = local:%var_dir/.daemon
    ```
  - pid — real address of a process must be read from a PID file of a process. This address type is allowed only in some cases, and in such a case this will be explicitly pointed out in parameter description.
- **text value, string** — parameter value is textual string (some text). Can be enclosed in quotation marks. If value contain white spaces, then it must be enclosed in quotation marks.
- **log level** — parameter value is string with name of verbosity level which is used for output messages in some log or **syslog** system service. List of allowed verbosity levels see in chapter Logging.
- **possible values** — parameter has the type which has not been described in the previous points of this list. In this case its allowed values are listed.

Behavior of modules in case of incorrect configuration files

- If any parameter is incorrect, respective module of **Dr.Web**

for Novell Storage Services would output error message to console (if module is running in foreground mode), log file and terminate.

- When any unknown parameter is found in configuration file, **Dr.Web for Novell Storage Services** modules continue execution and output warning to the log file.

Some parameters can use regular expressions (for each parameter it is noted in its description) as values. Syntax of regular expressions of **Perl** is by default used. You can familiarize with bases of regular expressions, for example, in **Wikipedia** (article Regular expressions).

Logging

All **Dr.Web for Novell Storage Services** components keep records about their operation in the logs. You can set a log mode for each component (output information into the file or to **syslog**).

You can also select a log verbosity level: for example, set high level of verbosity (the `Debug` option) or disable logging (the `Quiet` option). To set a verbosity level, use the **`LogLevel`** parameter. You can also specify additional parameters for certain modules to configure their verbosity log level (for example, keeping records of IPC subsystem operation is modified by the IPCLevel parameter).

If the **`LogLevel`** configuration parameter is not available for a module, it is not allowed to adjust its log mode. In this case, the default log mode has a verbosity level similar to `Debug`.

Log verbosity levels

If allowed, you can set one of the following log verbosity levels for a **Dr.Web for Novell Storage Services** component (the list is arranged in ascending order of detail):

- `Quiet` – Logging is disabled.
- `Error` – The component logs only fatal errors.
- `Alert` – The component logs errors and important warnings.
- `Warning` – The component logs errors and all warnings.
- `Info` – The component logs errors, warnings and information messages.
- `Notice` – This mode is similar to the `Info` mode, but the component also logs notifications.
- `Debug` – This mode is similar to the `Notice` mode, but the component also logs debug information.
- `Verbose` – The component logs all details on its activity (this mode is not recommended, because a large volume of logged data can considerably reduce performance of both the program and **syslog** service if it is enabled).

Each **Dr.Web for Novell Storage Services** module can have different set of allowed log verbosity levels. For information on available verbosity levels, see description of the corresponding parameters.

Logging into syslog

If you select the mode of logging information into **syslog**, it is necessary to specify a verbosity log level and a message source label. The label can be used by the **syslog** service for internal routing of messages to different logs. Routing rules are configured in the **syslog** daemon configuration file (usually, the path to the file is `/etc/syslogd.conf`).

To set a flag for syslog messages, specify the SyslogFacility parameter value in configuration files. You can specify one of the following parameter values:

- `Daemon` – a label of a resident system service (daemon) message;
- `Local0`, ..., `Local7` – a label of a user application message (8 values are reserved `Local0` to `Local7`);
- `Kern` – a label of a system kernel message;
- `User` – a label of a user process message;
- `Mail` – a label of a mail system message.

Note that if information is logged into **syslog**, an additional parameter **`SyslogPriority`** can be specified in configuration files. The **`SyslogPriority`** parameter defines a verbosity level of logging into **syslog** and is modified by one of the values available for the **`LogLevel`** parameter. If you select the mode of logging into the file, the **`SyslogPriority`** parameter is ignored. Otherwise, information is logged into **syslog** with a less verbosity level.

Example:

Let us assume that logging of a module is defined by the following parameter values: **`LogLevel`** = `Debug`, **`SyslogPriority`** = `Error`. If mode of logging into **syslog** is selected, the log verbosity level is `Error` (that means only records about errors are to be logged and the `Debug` value is ignored).

Allowed actions

You can configure **Dr.Web for Novell Storage Services** components to apply specified actions to objects that are detected to be malicious, suspicious or potentially dangerous.

You can use the following actions when configuring the settings:

You can use the following actions when configuring **Dr.Web Scanner**:

- `Move` – move the file to the **Quarantine** folder;
- `Delete` – delete the infected file;
- `Rename` – rename the file;
- `Ignore` – ignore the file;
- `Report` – only log information about the file;
- `Cure` – try to cure the infected object.

The following actions are available for **Dr.Web NSS**:

- `Pass` – ignore the file;
- `Cure` – try to cure the infected object;
- `Report` – only send the report to log;
- `Quarantine` – move the file to the **Quarantine** folder and restrict access to the object;
- `Remove` – delete the file.

Please note, that action names are case insensitive (for example, value `Report` is equal to `report`).

Installation and Deinstallation

Below you can find detailed description of **Dr.Web for Novell Storage Services** solution installation and deinstallation procedures for UNIX systems. Administrator (`root`) privileges are necessary to perform all these operations.

You must carefully uninstall all packages of earlier product versions from any previous installations.

Dr.Web for Novell Storage Services solution distribution package for UNIX systems is delivered in EPM format (script-based distribution package with installation and removal scripts and standard install/uninstall GUIs) designed to use with ESP Package Manager (EPM). Please note that all these scripts belong only to EPM-package itself, not to any of the components of **Dr.Web for Novell Storage Services**.

Installation, deinstallation and upgrade procedures for **Dr.Web for Novell Storage Services** solution can be carried out in the following ways:

- via install/uninstall GUIs;
- via install/uninstall console scripts.

During installation dependencies are supported, i.e. if for successful installation of any component some other components must be previously installed (e.g., `drweb-daemon` package requires **`drweb-common`** and **`drweb-bases`** packages to be previously installed), then they will be installed automatically.

If you install **Dr.Web for Novell Storage Services** solution to the computer, where some other **Dr.Web** products have been previously installed from EPM-packages, then at every attempt to remove some modules via uninstall GUI you will be prompted to remove absolutely all **Dr.Web** modules, including those from other products.

Please, pay special attention to the actions you perform and selections you make during deinstallation to avoid accidental removal of some useful components.

Installation from Distribution Package for UNIX systems

Dr.Web for Novell Storage Services solution is distributed as a self-extracting package `drweb-nss_[version]~linux_[processor_architecture].run`.

The following components are included into this distribution:

- `drweb-common`: contains main configuration file `drweb32.ini`, libraries, documentation and directory structure. During installation of this component `drweb` user and `drweb` group will be created;
- `drweb-bases`: contains Anti-virus search Engine (**Dr.Web Engine**) and virus databases. It requires `drweb-common` package to be previously installed;
- `drweb-libs`: contains common libraries for all the components of the software solution;
- `drweb-epm6.0.2-libs`: contains libraries for graphical installer and uninstaller. It requires `drweb-libs` package to be previously installed;
- `drweb-epm6.0.2-uninst`: contains files of graphical uninstaller. It requires `drweb-libs` package to be previously installed;
- `drweb-boost147`: containes common libraries for **Dr.Web Agent** and **Dr.Web Monitor**. It requires `drweb-libs` package to be previously installed;
- `drweb-updater`: contains update utility **Dr.Web Updater** for **Dr.Web Engine** and virus databases. It requires `drweb-common` and `drweb-libs` packages to be previously installed;

- `drweb-agent`: contains **Dr.Web Agent** executable files and its documentation. It requires `drweb-common` and `drweb-boost147` packages to be previously installed;
- `drweb-agent-es`: contains files required to run **Dr.Web Agent** in central protection mode. It requires `drweb-agent`, `drweb-updater` and `drweb-scanner` packages to be previously installed;
- `drweb-daemon`: contains **Dr.Web Daemon** executable files and its documentation. It requires `drweb-bases` and `drweb-libs` packages to be previously installed;
- `drweb-scanner`: contains **Dr.Web Scanner** executable files and its documentation. It requires `drweb-bases` and `drweb-libs` packages to be previously installed;
- `drweb-monitor`: contains **Dr.Web Monitor** executable files and its documentation. It requires `drweb-agent`, `drweb-common` and `drweb-boost147` packages to be previously installed;
- `drweb-perftools0`: contains **Google Performance Tools** library used by **Dr.Web NSS**. It requires `drweb-libs` package;
- `drwen-nss-doc`: contains **Dr.Web for Novell Storage Services** documentation;
- `drweb-nss`: contains **Dr.Web NSS** executable files. It requires `drweb-perftools0`, `drweb-agent` and `drweb-monitor` packages.

In distributions for 64-bit systems an additional package with libraries is included: `drweb-libs64` - containing libraries for 64-bit components.

To install all the components of **Dr.Web for Novell Storage Services** solution automatically you may use either console (CLI) or the default file manager of your GUI-based shell. In the first case allow the execution of the corresponding self-extracting package with the following command:

```
# chmod +x drweb-nss_[version]~linux_
[processor_architecture].run
```

and then run it:

```
# ./drweb-nss_[version]~linux_
[processor_architecture].run
```

As a result, `drweb-nss_[version]~linux_[processor_architecture]` directory will be created, and install GUI will be initialized. If startup has been performed without root privileges, install GUI will try to gain appropriate privileges by itself.

If install GUI fails to start, then interactive console installer will be initialized.

If you want only to extract the content of the package without starting install GUI, use `--noexec` command line parameter:

```
# ./drweb-nss_[version]~linux_
[processor_architecture].run --noexec
```

After you extract the content, you may initialize install GUI and continue setup with the following command:

```
# drweb-nss_[version]~linux_
[processor_architecture]/install.sh
```

To initialize console installer use the following command:

```
# drweb-nss_[version]~linux_
[processor_architecture]/setup.sh
```

During the installation the following processes take place:

- Original configuration files are recorded to the `%etc_dir/software/conf/` directory with the following names: `[configuration_file_name].N`.
- Operational copies of configuration files are placed to the corresponding directories of the installing software.
- Other files are installed. If in the corresponding directory file with the same name already exists (e.g. after inaccurate

removal of previous versions of the packages), it will be overwritten with the new file, and its copy will be saved as `[file_name].O`. If some `[file_name].O` file already exists in this directory, it will be replaced with the new file of the same name.

Using GUI Installer

To install with GUI

1. Execute the following command:

```
# drweb-nss_[version]~linux_
[processor_architecture]/install.sh
```

The setup program launches. On any step, click **Back** or **Next** to navigate, or click **Cancel** to abort installation.

On the Welcome screen, click **Next**.

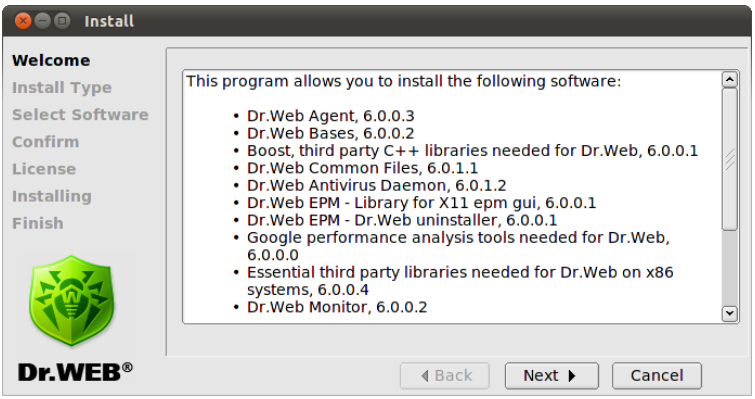


Figure 2. Welcome window

2. In **Install Type** window only one installation type is available: **Dr.Web for Novell Storage Services**. Click **Next** to continue installation.

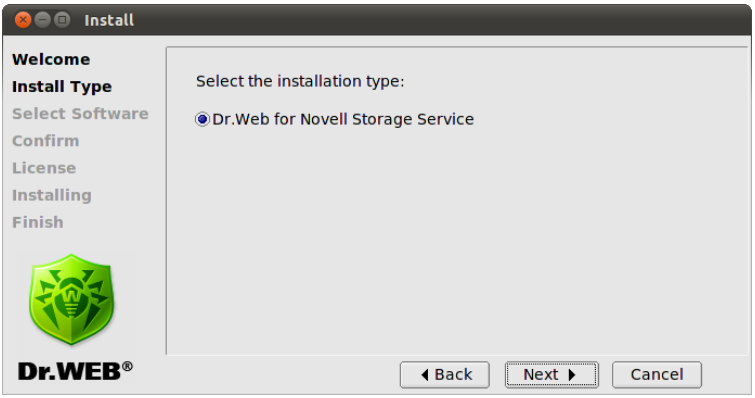


Figure 3. Install type window

Select necessary components on the **Select Software** screen:

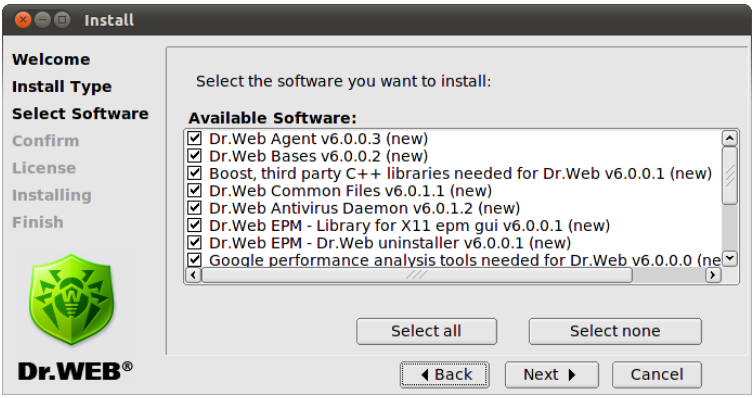


Figure 4. Select Software screen

If installation of a component requires some other components to be previously installed, all corresponding dependencies are selected for installation automatically. For example, if you select to install **Dr.Web Antivirus Daemon**, then **Dr.Web Bases** and **Dr.Web Common Files** are selected and installed automatically.

Click **Install None** to clear selection.

When you complete selection, click **Next**.

3. On the **Confirm** screen, review and confirm the list of components to install:

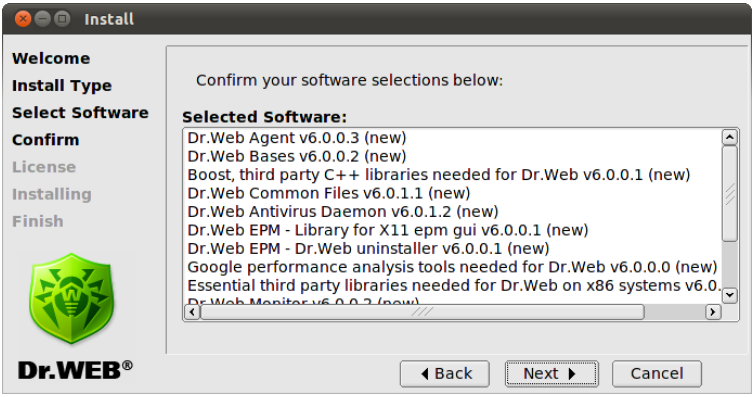


Figure 5. Confirm screen

Click **Next** to confirm selection, or click **Back** to make changes.

4. Review the **License Agreement**. To proceed, you need to accept it. If necessary, use the **Language** list to select preferred language:

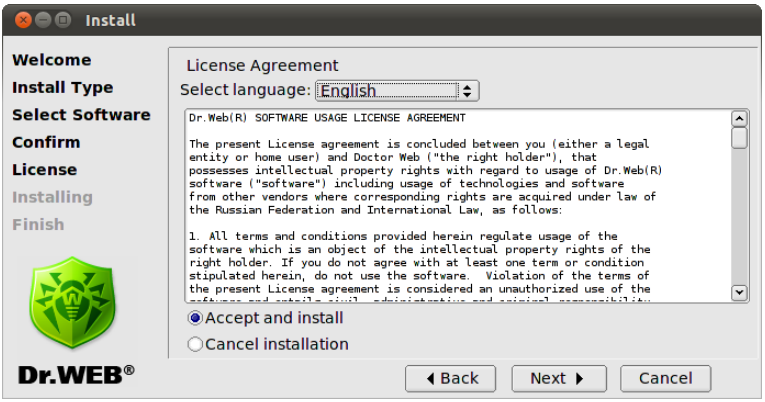


Figure 6. License Agreement screen

5. If you accepted the **License Agreement**, installation start. On the **Installing** screen, you can review the installation process in real-time:

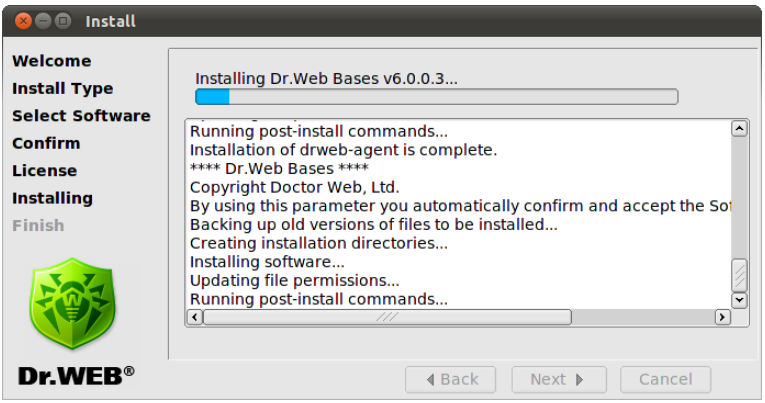


Figure 7. Installing screen

This report is logged at the same time in the `install.log` log file located at the `drweb-nss_[version]~linux_[processor_architecture]`

directory. If you selected to **Run interactive postinstall script**, then after installation of the components is completed, the post-install script is initialized for basic configuration of **Dr.Web for Novell Storage Services**.

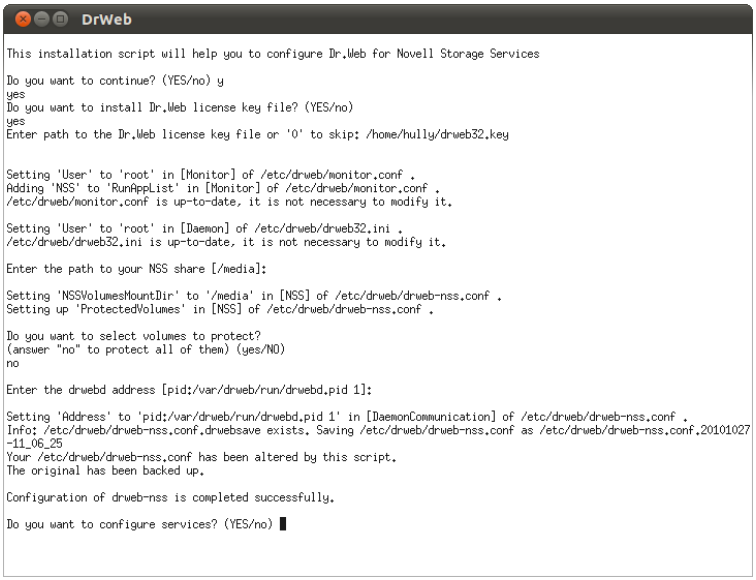


Figure. 8. Interactive post-install script

This script offers you:

- To install license key file, which you receive after product registration;
- To specify path to folder in where was mounted NSS partitions (NSS share);
- To specify, if it is necessary, which NSS partitions will be protected from viruses (by default will be protected all partitions);

- To specify socket address for interaction with **Dr.Web Daemon** (drwebd address). By default will be offered real address (PID) of **Dr.Web Daemon** process, which is started on local host `pid:/var/drweb/run/drwebd.pid`;
- To start modules **Dr.Web Daemon** and **Dr.Web Monitor**, if license key file was installed (configure services).

If configuration files already exist, before modification their backup copies with the `.drwebsave` extension will be created.

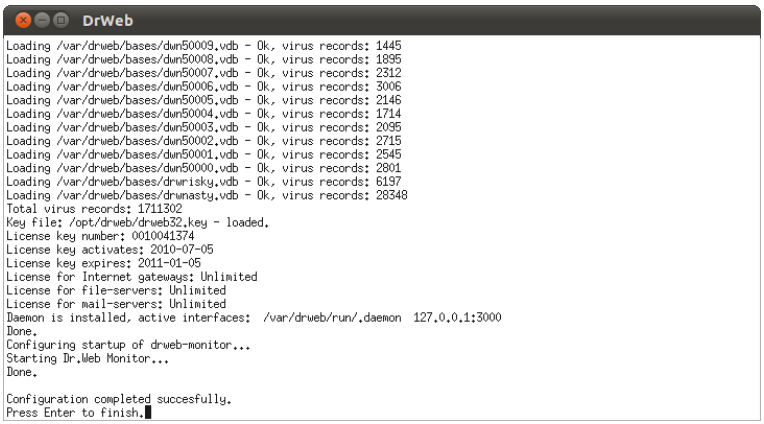


Figure 9. Starting services

On the **Finish** screen, click **Close** to exit setup:

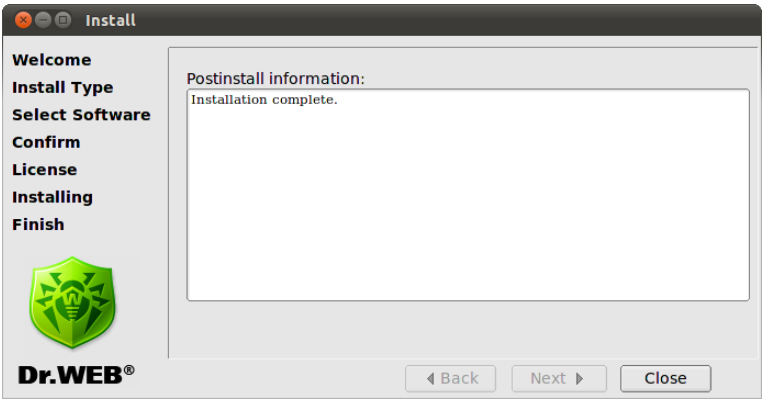


Figure 10. Finishing screen

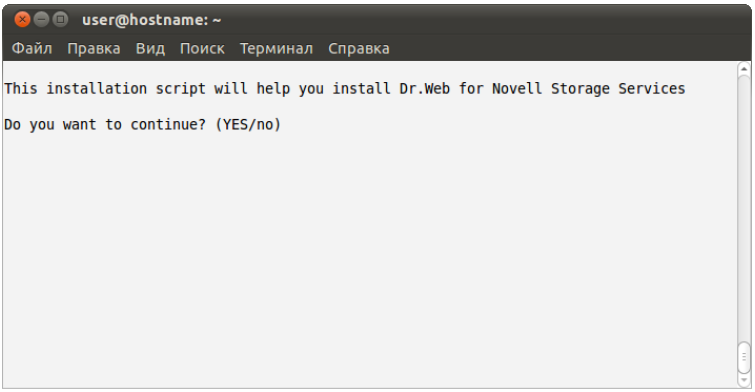
Using Console Installer

Console installer starts automatically is graphical installer fails to start. If console installer also fails to initialize (e.g., when it is impossible to gain necessary privileges), then you can try to run the following command with `root` privileges:

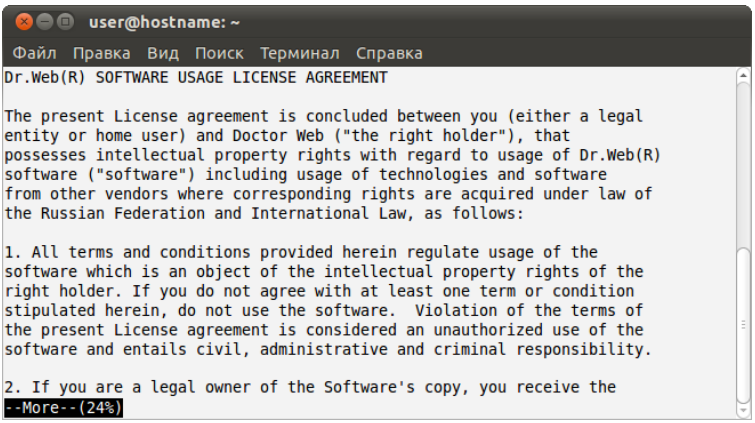
```
# drweb-nss_[version]~linux_
[processor_architecture]/setup.sh
```

To install from console

1. Once the console installer start, a conversation window opens:

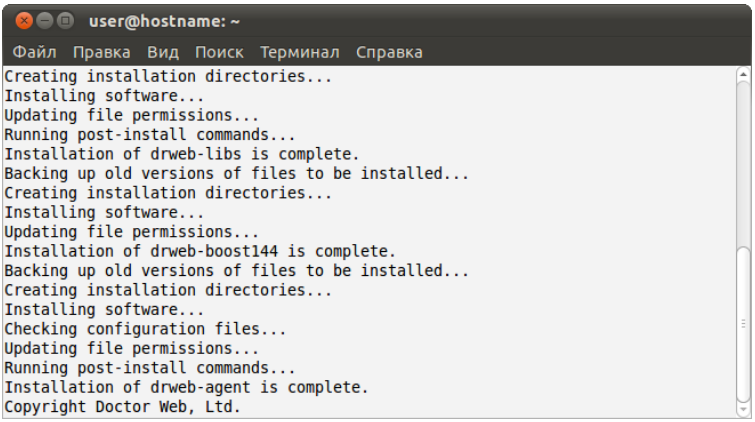


2. If you want to install **Dr.Web for Novell Storage Services,** type **Y** or **Yes** (values are case insensitive), otherwise type **N** or **No**. Press ENTER.
3. Review the **License Agreement**. To scroll the text, press SPACEBAR:

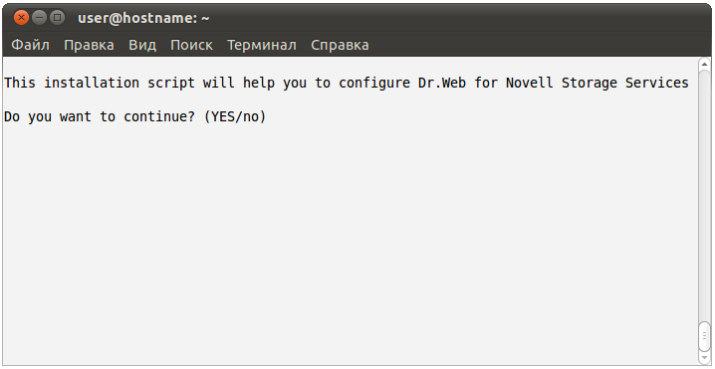


To continue the installation, you need to accept the **License Agreement**. If you agree to the terms, type **Y** or **Yes**.

4. The installation process starts immediately. You can review the installation process in console in real-time:



5. After installation of the components, the post-install script runs automatically to set up basic configuration of **Dr.Web for Novell Storage Services**. This script offers you to specify path to the license key file and enable automatically all the services necessary for proper operation of **Dr.Web for Novell Storage Services** (i.e., **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**). In additional script offers to you to specify path to a directory, in which mounted NSS partitions and to select which NSS partitions will be protected from viruses (by default will be protected all partitions).



Removal of Distribution Package for UNIX Systems

To remove all the components of **Dr.Web for Novell Storage Services** solution via uninstall GUI, initialize it with the following command:

```
# %bin_dir/remove.sh
```

If startup has been performed without root privileges, uninstall GUI will try to gain appropriate privileges by itself.

If uninstall GUI fail to start, then interactive console uninstaller will be initialized.

After deinstallation you can also remove `drweb` user and `drweb` group from your system.

During the deinstallation the following actions are performed:

- Original configuration files are removed from the `%etc_dir/software/conf/` directory.
- If operational copies of configuration files were not modified by the user, they are also removed. If the user has made any changes to them, they are preserved.
- Other **Dr.Web** files are removed. If a copy of some old file has been created at installation, this file will be restored under the name it had before the installation. Usually, such copies are named `[file_name].O`.
- License key files and log files are preserved in corresponding directories.

Using GUI Uninstaller

To uninstall with GUI

1. Execute the following command:

```
# %bin_dir/remove.sh
```

The setup program launches. On any step, click **Back** or **Next** to navigate, or click **Cancel** to abort installation.

On the Welcome screen, click **Next**:

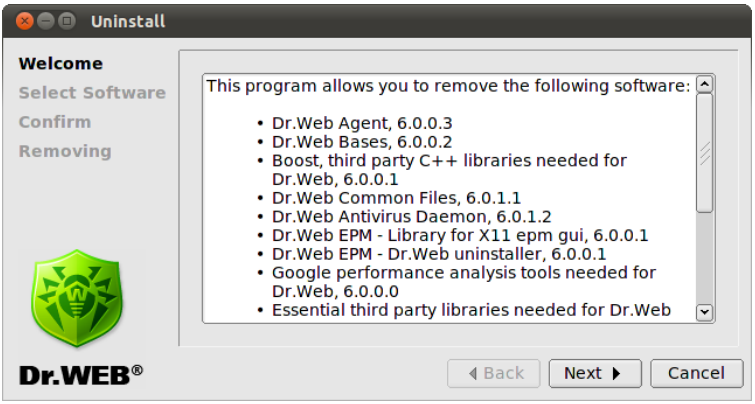


Figure 11. Welcome screen

2. On the **Select Software** screen, select components to remove:

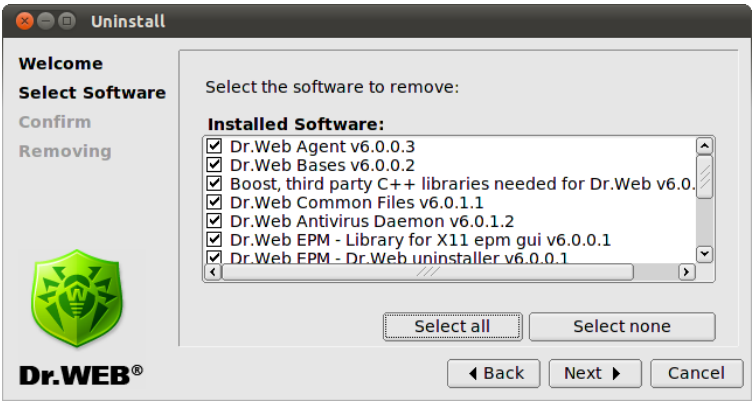


Figure 12. Select Software screen

All corresponding dependencies will be selected for de-installation automatically.

If you installed **Dr.Web for Novell Storage Services** solution on a computer with another **Dr.Web** product installed from EPM-packages, then setup lists all **Dr.Web** modules for both **Dr.Web for Novell Storage Services** and the old product. Please, pay attention to the actions you perform and selections you make during de-installation to avoid accidental removal of useful components.

Click **Remove All** to select all components, or click **Remove None** to clear selection.

When you complete selection, click **Next**.

3. On the **Confirm** screen, review and confirm the list of components to remove:

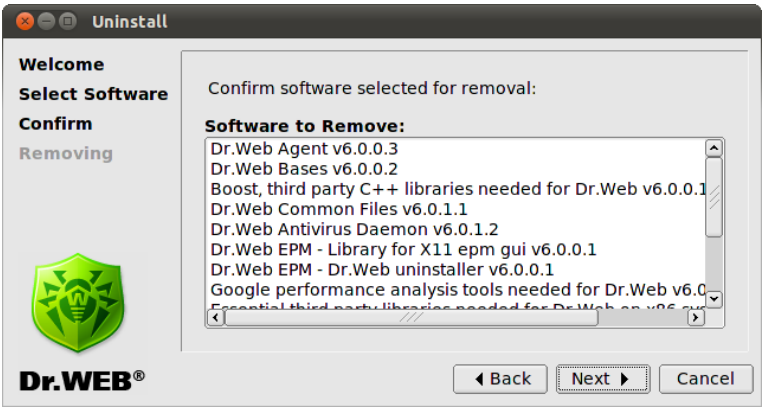


Figure 13. Confirm screen

Click **Next** to confirm selection, or click **Back** to make changes.

4. On the **Removal** screen, you can review removal process in real-time:

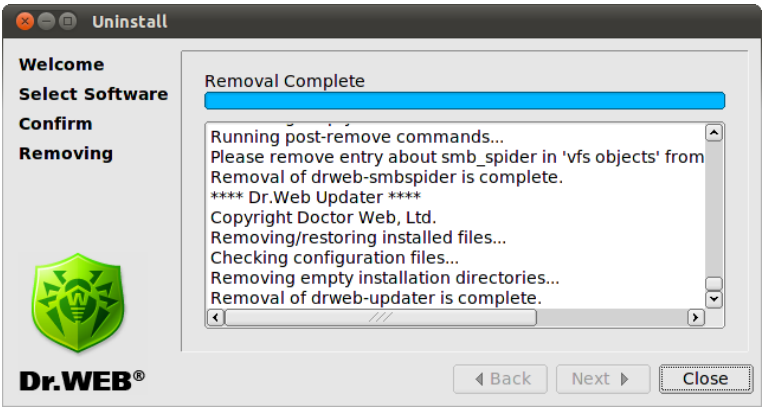


Figure 14. Removal screen

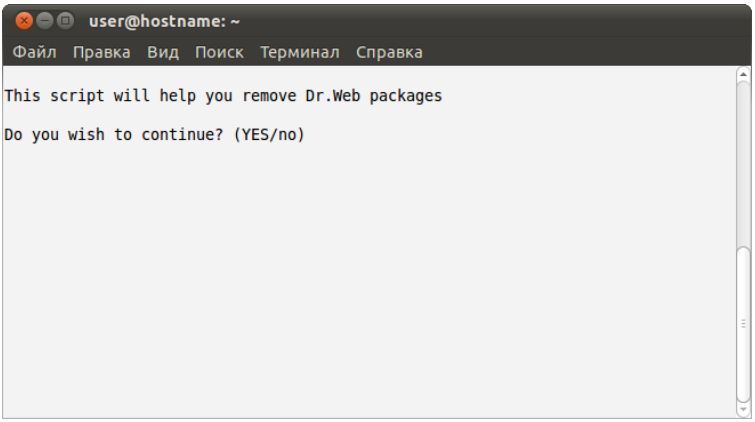
5. Click **Close** to exit setup.

Using Console Uninstaller

Console uninstaller starts automatically when graphical uninstaller fails to start.

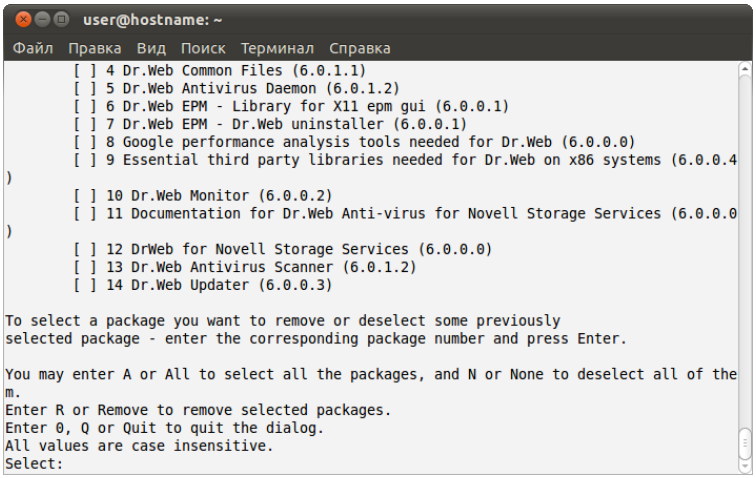
To uninstall from console

1. Once the console uninstaller start, a conversation window opens:

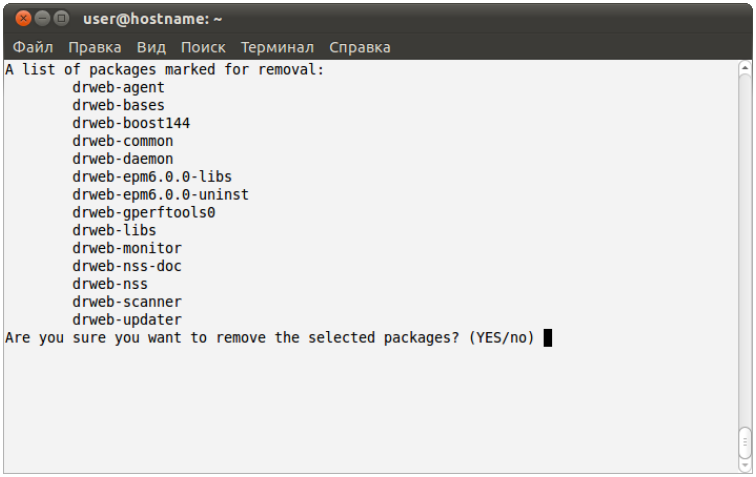


   If you want to uinstall **Dr.Web for Novell Storage Services**, type **yes**, otherwise type **no**. Press ENTER.

2. Review the list of components available for removal:



3. Follow the prompts to select components to remove.
4. To start uninstall, confirm you selection by typing **Y** or **Yes** and pressing ENTER (values are case insensitive):



5. You can review removal process in console in real-time.
6. Once the process completes, exit setup.

Upgrade of Distribution Package for UNIX Systems

Upgrade process combines install and uninstall procedures. If you want to upgrade **Dr.Web for Novell Storage Services** solution, you must download the latest version of corresponding software, remove the previous version and install the new one.

After you upgrade **Dr.Web for Novell Storage Services** solution, license key files, log files and configuration files, modified by the user, are preserved in corresponding directories.

Startup of Dr.Web for Novell Storage Services

You can run **Dr.Web for Novell Storage Services** and perform its initial configuration using interactive configuration script.

If you want to manually run **Dr.Web for Novell Storage Services**

1. Register the software.
2. Place the key file `drweb32.key` to the directory with **Dr.Web** executable files (default directory is `/opt/drweb/`). If you want to use key file from the different location, you must specify full path to it as a **`Key`** parameter value of main configuration file `drweb32.ini`. Because **Dr.Web for Novell Storage Services** can operate only in `Standalone` mode (without integration with **Dr.Web Enterprise Security Suite**), path to the key file must also be set as a value of **`LicenseFile`** parameter in **Dr.Web Control Agent** configuration file `agent.conf`.
3. Configure the software by making necessary changes in configuration files. Please refer to the corresponding chapters of this Manual for the detailed information on configuration.
4. Open `drwebd.enable` file and set the value of `ENABLE` variable to `1` in order to enable running of **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** locally (properly configured and working **Daemon** on some other computer in the network is used), `ENABLE` value must be set to `0`.
5. Open `drweb-monitor.enable` file and set the value of `ENABLE` variable to 1 in order to enable running of **Dr.Web Monitor**.
6. Run **Dr.Web Daemon** and **Dr.Web Monitor** using console or your file manager. After startup **Dr.Web Monitor** will initialize all other modules of **Dr.Web for Novell Storage Services** solution. You may also run each module independently, but **Dr.Web Control Agent** module must be ran first, since all other modules receive configuration information from **Agent**.

Placement of the `enable` files is depends from **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:

  Files will be placed to directory `%etc_dir` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

- Installation from **native DEB packages**:

  Files will be placed to directory `/etc/defaults` and will be named
  `drwebd`,
  `drweb-monitor`.

- Installation from **native RPM packages**:

  Files will be placed to directory `/etc/sysconfig` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

For Linux and Solaris

To run the **Dr.Web for Novell Storage Services** solution, do the following:

1. Register the software.
2. Place the key file to the directory for **Dr.Web for Novell Storage Services** executable files (default directory for UNIX systems is `%bin_dir`). Key file name may vary depending on the distribution kit (for the detailed information, see Software Registration chapter):

- If **Dr.Web for Novell Storage Services** was purchased as a standalone product License key file is called `drweb32.key`. In this case, you should just copy file to `%bin_dir` directory without changing its name.
- If **Dr.Web for Novell Storage Services** was purchased as a part of **Dr.Web Enterprise Security Suite** set, archive received during registration contains a key file for the **Enterprise Server** (`enterprise.key`) and a key file for workstations (`agent.key`). Rename `agent.key` as `drweb32.key` and copy it to the `%bin_dir` folder.

If you want to use key file with a different location or name (for example, `agent.key`), you must specify its full path as a `Key` parameter value of configuration file `drweb32.ini`. While working in `Standalone` mode, this alternative path to the key file must be also specified in the value of **`LicenseFile`** parameter of the `agent.conf` configuration file of the **Dr.Web Agent** component.

3. Configure the software by making necessary changes to configuration files. Refer to the corresponding chapters of this Manual for the detailed information on configuration.
4. In `drwebd.enable` file set `1` as a value of `ENABLE` variable to enable startup of **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** (properly configured and working **Daemon** on some other computer in the network is used), `ENABLE` value must be set to `0` (it is also used as a default value).
5. In `drweb-monitor.enable` file set `1` as a value of `ENABLE` variable to enable startup of **Dr.Web Monitor**.

Placement of the `enable` files is depends from **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:

  Files will be placed to directory `%etc_dir` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

- Installation from **native DEB packages**:

  Files will be placed to directory `/etc/defaults` and will be named
  `drwebd`,
  `drweb-monitor`.

- Installation from **native RPM packages**:

  Files will be placed to directory `/etc/sysconfig` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

6. Start initializing scripts for **Dr.Web Daemon** and **Dr.Web Monitor** either from console or from any file manager of your operation system. After startup **Dr.Web Monitor** will initialize all other components of the **Dr.Web for Novell Storage Services** solution. Also each component can be started independently, but **Dr.Web Agent** module must be started first, because all other components will receive their configuration through the **Agent**.

   In case of installation from native packages in Solaris:

   Through **Dr.Web for Novell Storage Services** installing, service management system SMF attempts to launch **Dr.Web Monitor** component. If **Monitor** can't find licence key file (for example in case of first **Dr.Web for Novell Storage Services** installing), it stops it's work and changes SMF to maintenance state.

To launch **Monitor**, maintenance state should be reseted:

- Enter the command

```
# svcs -p <FMRI>
```

  where FMRI - unique identifier of controlled resource, in this case - **Dr.Web Monitor** component.

- Forcibly cancel processes from `svcs -p` output list.

```
# pkill -9 <PID>
```

  where PID — number of process, that listed above.

- Restart **Dr.Web Monitor** with command

```
# svcadm clear <FMRI>
```

While installing **Dr.Web for Novell Storage Services** from native packages in Solaris, complex launches with service management system SMF:

```
# svcadm enable <drweb-monitor>
# svcadm enable <drweb-daemon>
```

To stop service enter:

```
# svcadm disable <service_name>
```

Module `drwebd` can be launched in two modes:

1. Standard run through the `init` script
2. Using **Dr.Web Monitor**

While working in second mode, you need to set `ENABLE` parameter to 0 in `enable` file.

Each of the components can be started manually as well, but note, that **Dr.Web Agent** component must be initialized beforehand in order to provide configuration information to all the other components.

For FreeBSD

To run the **Dr.Web for Novell Storage Services** solution, do the following:

1. Register the software.
2. Place the key file to the directory for **Dr.Web for Novell Storage Services** executable files (default directory for UNIX systems is `%bin_dir`). Key file name may vary depending on the distribution kit (for the detailed information, see Software Registration chapter):
   - If **Dr.Web for Novell Storage Services** was purchased as a standalone product License key file is called `drweb32.key`. In this case, you should just copy file to `%bin_dir` folder without changing its name.
   - If **Dr.Web for Novell Storage Services** was purchased as a part of Dr.Web Enterprise Security Suite set, archive received during registration contains a key file for the **Enterprise Server** (`enterprise.key`) and a key file for workstations (`agent.key`). Rename `agent.key` as `drweb32.key` and copy it to the `%bin_dir` directory.

   If you want to use key file with a different location or name (for example, `agent.key`), you must specify its full path as a `Key` parameter value of configuration file `drweb32.ini`. While working in `Standalone` mode, this alternative path to the key file must be also specified in the value of **`LicenseFile`** parameter of the `agent.conf` configuration file of the **Dr.Web Agent** component.
3. Configure the software by making necessary changes to configuration files. Refer to the corresponding chapters of this Manual for the detailed information on configuration.
4. Add the following lines to the `/etc/rc.conf` file:

- `drwebd_enable="YES"` – to enable startup of **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** (properly configured and working **Daemon** on some other computer in the network is used), then you can just not include the specified line in the `rc.conf` file;
- `drweb_monitor_enable="YES"` – to enable startup of **Dr.Web Monitor.**

5. Start initializing scripts for **Dr.Web Daemon** and **Dr.Web Monitor** either from console or from any file manager of your operation system. After startup **Dr.Web Monitor** will initialize all other components of the **Dr.Web for Novell Storage Services** solution. Also each component can be started independently, but **Dr.Web Agent** module must be started first, because all other components will receive their configuration through the **Agent**.

Each of the components can be started manually as well, but note, that **Dr.Web Agent** component must be initialized beforehand in order to provide configuration information to all the other components

OS with SELinux

To set up successful operation of **Dr.Web Scanner** and **Dr.Web Daemon** components in OS protected by **SELinux**, you must compile politics for operation with corresponding modules **drweb-scanner** and **drweb-daemon** or set 1 as a value of allow_execheap variable.

Templates used in compilation of modules for politics may vary widely, depending on the type of **Linux** distribution, its version, set of **SELinux** politics and user settings. To receive more detailed information on compilation of politics you may refer to corresponding documentation on your **Linux** distribution.

To create necessary politics :

1. Create new **SELinux** policy source file (.te file). This file define the access rules related to described module. You can create necessary politics:
   - Using **policygentool** utility. To do this, specify two parameters: the name of the policy module (interaction with which has to be adjusted) and the full path to the corresponding executable.

     Please note that **policygentool** utility which included in `selinux-policy` package in **Red Hat Enterprise Linux** and **CentOS Linux**, may not work correctly. In this case, use utility **audit2allow**.

     Example:

     For **Dr.Web Scanner**:

     ```
     # policygentool drweb-scanner /opt/drweb/
     drweb.real
     ```

     For **Dr.Web Daemon**:

     ```
     # policygentool drweb-daemon /opt/drweb/
     drwebd.real
     ```

     You will be prompted to enter a few common domain characteristics, and for each module three files will be created: `[module_name].te`, `[module_name].fc` and `[module_name].if`.
   - Using **audit2allow** utility. This utility generates policy modules based on reports of denial of access from system log files. Reports can be searched automatically in system log files or you can set the path to log file manually.

In general, when using the `audit` daemon, audit log located in `/var/log/audit/audit.log` file. Otherwise, AVC messages are stored in `/var/log/messages` log file.

`audit2allow` utility is included in `policycoreutils-python` package (for **RedHat Enterprise Linux**, **CentOS**, **Fedora**) or in `python-sepolgen` package (for **Debian**, **Ubuntu**).

Example:

```
# audit2allow -M -i /var/log/audit/audit.
log drweb
```

In this example, **`audit2allow`** search AVC messages in `audit.log` file.

Example:

```
# audit2allow -a -M drweb
```

In this example, **`audit2allow`** search AVC messages in system log files automaticaly.

In both cases, **`audit2allow`** creates two files: **SELinux** source file of policy (`drweb.te`) and compiled policy module `drweb.pp`. If you want to make changes to the access rules of **Dr.Web for Novell Storage Services** components, then edit `drweb.te` and go to step 2. If you don't want to change policy file, go to step 4 to install `drweb.pp` policy module.

2. Using **`checkmodule`** utility, create a binary representation (.mod file) of the policy source file. Please note that for successful policy compilation a `checkpolicy` package must be installed on the system.

Example:

```
# checkmodule -M -m -o drweb.mod drweb.
te
```

3. Create policy module (`drweb.pp`) by using **`semodule_package`** utility.

   Example:

   ```
   # semodule_package -o drweb.pp -m drweb.mod
   ```

4. To install the new policy module into the module store, use the **`semodule`** utility.

   Example:

   ```
   # semodule -i drweb.pp
   ```

It is also possible (but **not recommended!**) to set `1` as a value of `allow_execheap` environment variable to set up operation of **Dr.Web Scanner** and **Dr.Web Daemon** in **SELinux**. `allow_execheap` variable allow or deny execution of data in memory heap for all applications that runs in *unconfined domain*.

To set value of `allow_execheap` variable, execute the following command:

```
# setsebool -P allow_execheap=1
```

Software Registration. License Key File

User privileges for using **Dr.Web for Novell Storage Services** solution are controlled by special file called license key file.

License key file contains the following information:

- list of **Dr.Web for Novell Storage Services** components licensed to user;
- license expiration date;
- other restrictions (for example, number of protected workstations).

License key file has `*.key` extension and by default must be placed in a directory for **Dr.Web for Novell Storage Services** executable files.

License key file is digitally signed to prevent its editing. Edited license key file becomes invalid. It is not recommended to open your license key file in text editors to avoid its accidental corruption.

Users who have purchased **Dr.Web for Novell Storage Services** solution from **Doctor Web** certified partners obtain the license key file. The parameters of the key file are specified according to the license user has paid for. The license key file contains the name of the user (or a company name), and the name of the selling company.

For evaluation purposes users may also obtain a demo key file. It allows user to enjoy full functionality of the **Dr.Web for Novell Storage Services** solution, but has a limited term of use, and no technical support is provided.

License key file may be supplied as:

- a `drweb32.key` file license key for workstations, or as a zip archive containing license key file in case of purchasing **Dr. Web for Novell Storage Services** as a standalone product;
- a zip-archive, which contains a key file for the Server

(enterprise.key) and a key file for workstations (agent.key) in case of purchasing **Dr.Web for Novell Storage Services** as a part of **Dr.Web Enterprise Security Suite**.

License key file may be received using one of the following ways:

- sent by e-mail as a ZIP-archive containing license key file with *.key extension (usually after registration on the web site). Extract license key file using the appropriate archiving utility and copy/move it to the directory for **Dr.Web for Novell Storage Services** executable files (default directory for UNIX systems is %bin_dir);
- included into the distribution package;
- supplied on a separate media as a file with *.key extension. In this case user must copy it manually to the %bin_dir directory.

License key file is sent to user via e-mail usually after registration on the web site (web site location is specified in registration card accompanying the product). Visit the site, fill in the web form with your customer data and submit your registration serial number (printed on the registration card). As a result of this procedure license is activated, and license key file is created for the serial number provided. Then it is sent to user on the e-mail address specified.

It is recommended to keep license key file until it expires, and use it when reinstalling or repairing **Dr.Web for Novell Storage Services** solution installation. If the license key file is damaged or lost, it can be recovered by the same procedure as during license activation. In this case you must use the same product serial number and customer data you have entered during the registration, only e-mail address can be changed (in this case license key file will be sent to the new e-mail address). If serial number matches any entry in **Dr.Web for Novell Storage Services** database, the corresponding key file will be dispatched to user by automatic system using e-mail address provided.

Registration with the same product serial number can be performed up to 25 times. If you need to recover lost license key file after 25th registration, you must make a request for license key file

recovery on http://support.drweb.com/request/, and also specify all data used during registration, valid e-mail address and detailed description of the situation. Request will be considered by **Dr.Web for Novell Storage Services** technical support service engineers, and after approval license key file will be provided to user via automatic support system or dispatched via e-mail.

Path to license key file of the certain component must be specified as a **Key** parameter value in corresponding configuration file (`drweb32.ini`).

Example:

```
Key = %bin_dir/drweb32.key
```

If license key file specified as a **Key** parameter value is failed to read (wrong path, permission denied), expired, blocked or invalid, the corresponding component terminates.

When less than two weeks left until the license expiration, **Dr.Web Scanner** outputs warning message at start and **Dr.Web Daemon** notifies user via e-mail. Messages are sent at every startup, restart or reload of the **Daemon** for every license key file installed. To enable this option you must set up **MailCommand** parameter in `[Daemon]` section of `drweb32.ini` configuration file.

If you want to use key file from the different location, you must specify full path to it in the value of **LicenseFile** parameter from the `[StandaloneMode]` of the **Dr.Web Agent** configuration file (refer to the [StandaloneMode] Section description).

Dr.Web for Novell Storage Services

Dr.Web for Novell Storage Services protects NSS volumes using the following interacting modules:

- **Dr.Web NSS** – resident module used for integration with NSS file system;
- **Dr.Web Daemon** – resident module used for checking files for viruses and other malware;
- **Dr.Web Monitor** – utility module used for running, restarting and terminating **Dr.Web** modules in the necessary order;
- **Dr.Web Agent** gathers statistics, manages configuration of other **Dr.Web** modules and provides integration with **Dr.Web Enterprise Security Suite**.

Dr.Web NSS monitors file system changes in chosen NSS volumes and processes modified files according to its settings. NSS volumes to be monitored are specified in the `[NSS]` section of configuration file (`drweb-nss.conf`):

- If **`ProtectedVolumes`** configuration parameter is specified, **Dr.Web NSS** will monitor volumes specified in that parameter.
- If **`ProtectedVolumes`** configuration parameter is not specified, **Dr.Web NSS** will monitor all volumes mounted in directory specified in **`NSSVolumesMountDir`** configuration parameter.

Files undergo preliminary checkup before being sent to **Dr.Web Daemon** for scanning. Following files will not be scanned:

- files with zero size;
- files with size greater than value of **`MaxFileSizeToScan`** parameter in `[NSS]` section of configuration file (only if that value is not zero);
- files at paths specified in **`ExcludedPaths`** parameter in `[NSS]` section of configuration file. Files at paths specified in **`IncludedPaths`** parameter will always be scanned

regardless of **ExcludedPaths** parameter.

Files that pass preliminary checkup are added to the internal queue for scanning. On receiving HUP signal **Dr.Web NSS** will output list of tasks in the queue to the log file at INFO **syslog** verbosity level. Tasks in queue are processed by the thread pool which can be configured in **CheckPoolOptions** parameter in the [NSS] section of configuration file.

Files to be checked for viruses are sent to the **Dr.Web Daemon** for scanning. You can configure interaction with **Dr.Web Daemon** in [DaemonCommunication] section of configuration file. **Dr.Web NSS** is able to work with **Dr.Web Daemon** running locally and with **Dr.Web Daemons** on other machines via sockets. You can specify socket addresses and their weights in **Address** parameter in the [DaemonCommunication] section of configuration file. Weights are used to distribute load on servers running **Dr.Web Daemons**: addresses with higher weighs will be getting more scanning requests.

If any threats have been found by **Dr.Web Daemon**, compromised files are processed according to parameters in [Actions] section of configuration file depending on the type of threat.

Files with threats can be deleted or moved to **Quarantine** (settings of the **Quarantine** are located in the [Quarantine] section of configuration file). Notification to be sent upon detection of threat can be configured in [Notifications] section of configuration file. Scanning information for each file is written to the log file (logging settings are located in the [Logging] section of configuration file).

Information on every scanned file is sent to **Dr.Web Agent** for gathering statistics. Information on detected threats is sent immediately, information on scanned files is sent periodically (period is specified in **SendPeriod** parameter). Statistics gathering can be configured in [Stat] section of configuration file.

If errors have emerged during file processing, **Dr.Web NSS** will attempt to perform an action specified in the value of **ProcessingError** parameter from [Actions] section of

configuration file.

Command Line Parameters

As any UNIX program, **Dr.Web NSS** solution support command line parameters. You can use following command to run **Dr.Web NSS**:

```
drweb-nss [<parameters>] <agent_socket>
```

where:

- `parameters` are optional command line parameters;
- `agent_socket` is the socket through which **Dr.Web for Novell Storage Services** modules receive configuration information from **Dr.Web Agent**.

In the current version of **Dr.Web NSS** module support the following command line parameters:

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and exit		
`-v`	`--version`	
Description: Show current module version on the screen and exit		
`-l`	`--level`	`<level>`
Description: Set the log verbosity level for **Dr.Web NSS** module. Default value is info		
`-t`	`--timeout`	`<value in seconds>`
Description: Set the maximum waiting period for receiving configuration information from **Dr.Web Agent**		
	`--component`	`<name>`

Short case	Extended case	Arguments
Description: Set the name to be used in requests to **Dr.Web Agent** for configuration information		
	`--log-name`	`<name>`
Description: Set the name of the component to be used for logging		
	`--check-only`	
Description: Allows perform the configuration check. For proper functioning **Dr.Web Agent** must be running before. If check turns out to be successful, is output to console the message `Options OK` and otherwise – `Options ERROR`		

Example:

```
drweb-nss -t 30 local:/var/drweb/ipc/.agent
```

It runs **Dr.Web NSS** with 30 second timeout for receiving configuration data and with the **Dr.Web Agent** socket `local:/var/drweb/ipc/.agent`.

Signals

All background modules in **Dr.Web for Novell Storage Services** solution can process the following signals:

- `SIGHUP` – forces modules to reread their configuration files. When **Dr.Web Monitor** receives this signal, it makes all the running components reread their configuration.
- `SIGINT` and `SIGTERM` – after receiving any of these signals, modules finish their operation.

Dr.Web NSS can process additional signals:

- `SIGUSR1` – on receiving of this signal **Dr.Web NSS** saves thread pool and persistent connections statistics to the directory specified as a value of **`BaseDir`** parameter in the `[General]` section of configuration file.

- `SIGALRM` – on receiving this signal **Dr.Web NSS** sends all collected statistics to **Dr.Web Agent**.

Internal Statistics

Statistics on of thread pool and its persistent connections linked to these pools is collected only when it is enabled explicitly in thread pool settings (**`CheckPoolOptions`** parameter in the `[NSS]` section of **Dr.Web NSS** configuration file) by specifying an additional parameter **`stat`** `= yes`.

Example:

```
CheckPoolOptions = 2-20, stat = yes
```

Names of files created on receiving `SIGUSR1` signal look like the following:

- `name_(cli|srv)[.unique-id].txt` – for statistics on connections;
- `name_(thr[N])[.unique-id].txt` – for statistics on pools.

where:

- `name` – name of the mofule without `drweb-`.
- `cli` – for **client** connections.
- `srv` – for **server** connections.
- `unique-id` – for modules ran with unique identifier.
- `thr` – for thread pool.

If such file already exists, statistics will be added to the end of this file.

Each entry begins with the following:

```
=====================================================
======================
start:  Tue Oct 9 14:44:15 2008
curr:   Tue Oct 9 14:44:29 2008
period: 0d 0h 0m 14s
```

where date of the beginning of collecting statistics, current date and a period of time required for output are displayed.

For `srv` a number of created and closed connections and a maximum amount of elements in different queues are displayed.

```
closed: 0 (0 num/sec)
total created = 0 (0 num/sec)
max rea = 0 est = 0 don = 0 act = 0
```

For `cli` a number of connection created by request and closed on timeout, their average amount and current number are displayed.

```
created on request = 0 (0 num/sec)
closed by timeout = 0 (0 num/sec)
avg number = 0
current = 2
```

For `thr` output looks as follows:

```
min = 2 max = 2147483647 type = 0 freetime = 120
busy max = 0 avg = 0
requests for new threads = 0 (0 num/sec)
creating fails = 0
max processing time = 0 ms; avg = 0 ms
curr = 2 busy = 0
```

It contains:

- on the first line – maximum and minimum number of threads in one pool, type of the pool, maximum time (in seconds) for an additional thread to close if inactive;
- on the second line – maximum and average number of busy

threads;

- on the third line – number and frequency of requests for creation of additional threads;
- on the fourth line – number of failed attempts to create additional threads;
- on the fifth line – maximum and average time of processing of these requests;
- on the sixth line – current number of threads in a pool and a number of busy threads.

Quarantine

Quarantine is used for isolation of infected and suspicious files. If `quarantine` action is applied to a file, it is moved to quarantine directory. Path to this directory is specified in **`Path`** parameter in the section `[Quarantine]` of configuration file.

When file is moved to **Quarantine**, 6 random characters are appended to its name. In addition to this file, auxiliary file with service information (path to original location, original privileges and attributes) is created. Its name is the same as the modified name of the quarantined file with added postfix `-info`. Permissions for both files are set according to **`FilesMode`** parameter in the `[Quarantine]` section of configuration file.

Example:

`eicar.com` – original filename;

`eicar.comf8JRCG` – modified filename;

`eicar.comf8JRCG-info` – auxiliary file.

Some additional file properties supported by **NSS** (like quotas and NSS attributes) can be saved together with file in the **Quarantine**. This properties will be automatically reset to the file when it will be restored from the **Quarantine**. To enable this feature you must enable Linux extended attributes in **NSS** by adding the following lines to file `/etc/opt/novell/nss/nssstart.cfg`:

```
/ListXattrNWMetadata
/CtimeIsMetadataModTime
```

Please note that only **Open Enterprise Server 2** supports Linux extended attributes. To learn more about Linux extended attributes please refer to the Open Enterprise Server documentation.

Using drweb-nss-qcontrol

You can use **`drweb-nss-qcontrol`** utility to manage files and search in the **Quarantine** directory. At startup (if command line parameter `--agent` was not empty) it connects to the **Dr.Web Agent** to receive configuration information.

`drweb-nss-qcontrol` supports the following command line parameters:

- `-h [ --help ]` – displays information about available command line parameters;
- `-v [ --version ]` – display version number;
- `-l [ --level ] <level>` – sets log verbosity level (logging settings are determined by parameters in the `[Logging]` sections of configuration file, same as for **Dr.Web NSS**);
- `-i [ --ipc-level ] <level>` – sets log verbosity level for IPC library;
- `--log-filename <filename>` – sets log file name;
- `--agent <address>` – sets **Dr.Web Agent** address for receiving configuration information. If this parameters is not specified

- `--timeout <time>` – sets maximum waiting period for receiving **Dr.Web Daemon** replies and configuration information from **Dr.Web Agent**.
- `--show <regexp>` – displays general information on files in **Quarantine**. `<regexp>` specifies regular expression that matches all required filenames. Information is displayed in the following format:

```
NAME: original=[PATH] size=SIZE put_time=TIME
viruses=[VIRUSES] code=CODE mode=ATTRIBUTES
```

where:

- `NAME` – filename;
- `PATH` – full path to original file location;
- `SIZE` – file size in bytes;
- `TIME` – local time of moving file to **Quarantine**;
- `VIRUSES` – list of all viruses detected in the file;
- `CODE` – **Dr.Web Daemon** hexadecimal return code;
- `ATTRIBUTES` – original file attributes in octal form.

Saved **NSS** attributes are not displayed.

Example:

```
eicar.comf8JRCG: original=[/media/nss/VOLENC/
eicar.com]
size=105\put_time=2010-Aug-26 14:08:10
viruses=[infected with EICAR Test File\NOT a
Virus!)]
code=0x20 mode=0100666
```

- `--remove <regexp>` – removes from **Quarantine** directory files matching specified regular expression.

Example:

```
drweb-nss-qcontrol --remove .
```

All files in the **Quarantine** directory will be removed.

- `--restore <regexp>` – attempts to restore files matching

specified regular expression to their original location (or to another directory if `--restore-dir` command line parameter is specified). All file attributes will also be restored. File attributes supported only by **NSS** will be restored only if target directory is located in NSS volume.

If file to be restored is considered to be infected, you must specify path to in **ExcludedPaths** (and make sure that it is not specified in **IncludedPaths**) parameter in the `[NSS]` section of the configuration file. Otherwise **Dr.Web NSS** will immediately detect infected file and return it to the **Quarantine** (or process it in some other way according to its settings). If you want to restore suspicious files which have been determined to be clean after updating virus databases, you should use `--rescan` command line parameter. If some other file is present at the path where the file must be restored, you will be offered to overwrite it.

Example:

```
drweb-nss-qcontrol --restore eicar
```

Attempt to restore all files with names containing `eicar` at their original location.

- `--restore-dir <directory>` – sets restore directory used for `--restore` command line parameter.

  Example:

  ```
  drweb-nss-qcontrol --restore-dir sample/
  directory --restore eicar
  ```

  Attempt to restore all files with names containing '`eicar`' at specified directory "`sample/directory`".

- `--answer <answer>` – specifies answer to be used file overwriting confirmation when restoring files with `--restore` command line parameter.

  Example:

  ```
  drweb-nss-qcontrol --restore eicar --answer yes
  ```

  Attempt to restore all files with names containing `eicar` at their original location overwriting existing files automatically.

- `--rescan <regexp>` – sends all files with names matching regular expression to **Dr.Web Daemon** for rescanning. If file has been determined to be clear of threat during rescanning, it will be automatically restored.

  You can use this parameter to set up regular automatic restoration of cleared files in the **Quarantine**. Add the similar line to **crontab** (rescan file in the **Quarantine** every 30 minutes and restore cleared files without confirmation for overwrite):

  ```
  */30 * * * * sh -c "/opt/drweb/drweb-nss-
  qcontrol --rescan . --answer no"
  ```

Logging

Dr.Web for Novell Storage Services may log information using **syslog** daemon or to separate log file. In case of using **syslog**, messages are logged in the following format:

```
'['tid']' name[.sub] level text
```

where:

- `tid` – identifier of thread which sends the message;
- `name` – name of the corresponding module;
- `sub` – name of th module's service. The most important services are:
  - `ipc` – inter-process communication service;
  - `thrN` – thread pool support service, N is the number of the pool.
- `level` – log verbosity level. The following values can be used: `FATAL`, `ERROR`, `WARN`, `INFO`, `DEBUG`;
- `text` – text of the message written to the log file.

By default at startup every module sets `INFO` syslog level. After receiving configuration information from **Dr.Web Agent** level will be set according to its configuration.

If you need to set `DEBUG` syslog level at startup (for example, to get information about parameters received from **Dr.Web Agent**), use `--level` command line parameter.

Please note that after receiving configuration information from **Dr.Web Agent** level will be set according to its configuration, regardless of `--level` configuration parameter.

Checking Configuration

You can validate configuration files and configuration parameters received from **Dr.Web Agent**.To do so run a module with `--check-only` command line parameter. This parameter can be used only if **Dr.Web Agent** is running.

If all parameters are validated, following message will be output to the console:

```
Options OK
```

If any errors were found, their descriptions would be output to the console with message

```
Options ERROR
```

Dr.Web Monitor supports command line parameter `--check-all` to check **Dr.Web Monitor** configuration as well as configuration of all the modules it controls.

Configuration File

Dr.Web NSS settings are stored in the configuration file `/etc/drweb/drweb-nss.conf`.

Description of configuration file structure and parameter types can be found in the Configuration_Files chapter. Parameters are described in the order they are presented in configuration file.

[General] Section

Dr.Web NSS general settings are stored in the [General] section of configuration file.

Parameter	Description
[General] section	
BaseDir = {path to directory}	Main working directory. It contains sockets, databases and other files. In the current version value of this parameter can not be changed during system reload on SUGHUP signal. Default value: **BaseDir** = /var/drweb
MaxTimeoutForThreadActivity = {time}	Maximum time for a thread to close. This parameter is used at restart or shutdown of a system. Total amount of time for a system to sign off can be calculated in the following way: number of pools and the value of this parameter are multiplied, and then the time constant is added to the result. Default value: **MaxTimeoutForThreadActivity** = 2m
IpcTimeout = {time}	Timeout for establishing connection between components. Default value: **IpcTimeout** = 2m

[Logging] Section

[Logging] section stores logging settings. Logging is performed for all main modules of **Dr.Web for Novell Storage Services** solution.

Parameter	Description
[Logging] section	
Level = {log level}	Value of this parameter defines log verbosity level. The following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **Level** = Info
IpcLevel = {log level}	Log verbosity level of IPC library. The following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **IpcLevel** = Alert
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon

Parameter	Description
`FileName = {syslog | path to file}`	Path to log file name. You can specify `syslog` as log file name and logging will be carried out by `syslogd` system service. In this case you must also specify **`SyslogFacility`** parameter. Default value: **`FileName`** = `syslog`

[NSS] Section

This section contains settings for integration with **NSS** file system.

Parameter	Description
	`[NSS]`
`NSSVolumesMountDir = {path to directory}`	Path to directory where all NSS volumes are mount. You can specify which NSS volumes must be protected from viruses in `ProtectedVolumes` parameter. Default value: **`NSSVolumesMountDir`** = `/media/nss`
`ProtectedVolumes = {list of volumes}`	List of NSS volumes to be protected from viruses. If parameter value is empty, all subdirectories in **`NSSVolumesMountDir`** directory will be protected. If some subdirectories in that directory are not NSS volumes **Dr.Web NSS** will fail to initialize and output error message to the log file. Default value: **`ProtectedVolumes`** =

Parameter	Description
`CheckPoolOptions = {Pool Settings}`	Thread pool settings for processing scanning tasks. Default value: **`CheckPoolOptions`** = `{2-20}`
`HeuristicAnalysis = {logical}`	Enables or disables heuristic analysis. Heuristic analysis is capable to detect unknown viruses, which signatures are yet not included in virus databases. Since some benign programs can exhibit behaviour similar to viruses, heuristic analysis cannot determine whether file contains malware with absolute precision. All files detected by heuristic analyzer are considered suspicious. It s recommended to move such files to quarantine until new virus databases are released. You can send such files to **Doctor Web** company for analysis at http:/vms.drweb.com/sendvirus/. Using of heuristic analysis may increase scanning time. Default value: **`HeuristicAnalysis`** = `Yes`
`MaxFileSizeToScan = {Size}`	Maximum file size for scanning by **Dr.Web Daemon**. If file size is greater than this value, file will not be scanned. If parameter value is set to 0, file size will not be checked. Default value: **`MaxFileSizeToScan`** = `0b`
`IncludedPaths = {list of paths}`	List of relative paths which always will be protected from viruses. Files at this paths will always be scanned, regardless of **`ExcludedPaths`** parameter. All paths must be relative to directory

Parameter	Description
	specified in **NSSVolumesMountDir** parameter. All paths must be specified in normalized form (i.e. without symbols for current directory and parent directory: "." and ".."). Default value: **IncludedPaths** =
ExcludedPaths = {list of paths}	List of relative paths which will not be protected from viruses. Files at this paths will not be scanned, if path to them is not specified in **IncludedPaths** parameter. All paths must be relative to directory specified in **NSSVolumesMountDir** parameter. All paths must be specified in normalized form (i.e. without symbols for current directory and parent directory "." and ".."). Default value: **ExcludedPaths** =

[DaemonCommunication] Section

In this section you can configure **Dr.Web NSS** interaction with **Dr.Web Daemon**.

Parameter	Description
[DaemonCommunication]	
Address = {weighted addresses list}	Sockets used by **Dr.Web NSS** for interaction with **Dr.Web Daemon**(s). At least one valid address must be specified.

Parameter	Description

	Weighted addresses are specified in the following format:
	`ADDRESS WEIGHT`
	where `ADDRESS` is specified in standard socket address format (UNIX or TCP socket), and `WEIGHT` is an optional numeric value from a range of 0 to 100, defining priority of this address. Weight determines a relative load on a host in the network. Servers at sockets with greater wights will receive more requests.
	If the weight is set to 0, such addresses are considered backup addresses. Requests to backup addresses are sent only if there are no available addresses with weights equal to or greater than 1.
	Weight values should be assigned according to available resources on each server running **Dr.Web Daemon**.
	Examples:
	One address (path to pid file):
	`Address = pid:/var/drweb/run/drwebd.pid`
	Multiple addresses with assigned weights:
	`Address = pid:/var/drweb/run/drwebd.pid 10, \inet:3000@srv2.example.com 5`
	Default value:
	`Address = pid:/var/drweb/run/drwebd.pid 1`

`Timeout = {time}`	Maximum waiting period for **Dr.Web Daemon** command execution. If value is 0 (in any units), waiting period is unlimited.

Parameter	Description
	Default value: `Timeout = 2m`

[Actions] Section

In this section you can specify actions to be performed on detection of threats or errors during scanning and processing.

Available actions:

- `pass` – skip the file;
- `cure` – attempt to cure the infected file;
- `report` – only send notification and output message to the log file;
- `quarantine` – move file to the **Quarantine** directory;
- `remove` – remove the file.

Every action is logged to the log file. On all actions except `pass` notifications defined in [Notifications] section will be sent.

Parameter	Description
	`[Actions]`

`Infected = {action}`	Action to be performed in case the file is infected with a known virus.
	Allowed actions:
	`remove, quarantine, cure`
	Default value:
	`Infected = cure`

`Suspicious = {action}`	Action to be performed in case the file may be infected with an unknown virus. Allowed actions: `remove, quarantine, pass, report`

Parameter	Description
	Default value: **Suspicious** = quarantine
Incurable = {action}	Action to be performed in case the file is infected and cannot be cured (applicable only if **InfectedFiles** = Cure). Allowed actions: remove, quarantine Default value: **Incurable** = quarantine
Adware = {action}	Action to be performed in case the file contains adware program. Allowed actions: remove, quarantine, pass, report Default value: **Adware** = quarantine
Dialers = {action}	Action to be performed in case the file contains dialer program. Allowed actions: remove, quarantine, pass, report Default value: **Dialers** = quarantine
Jokes = {action}	Action to be performed in case the file contains joke program which can irritate or distract user. Allowed actions: remove, quarantine, pass, report Default value: **Jokes** = report

Parameter	Description
`Riskware = {action}`	Action to be performed in case the file contains riskware (potentially dangerous program). Allowed actions: `remove, quarantine, pass, report` Default value: **`Riskware`** `= report`
`ArchiveRestriction = {action}`	Action applied to archives which have not been scanned by **Dr.Web Daemon** due to restrictions specified in main configuration **`drweb32.ini`**. Allowed actions: `remove, quarantine, pass, report` Default value: **`ArchiveRestriction`** `= quarantine`
`Hacktools = {action}`	Action to be performed in case the file contains program used for hacking. Allowed actions: `remove, quarantine, pass, report` Default value: **`Hacktools`** `= report`
`SkipObject = {action}`	Action applied to files which cannot be scanned by **Dr.Web Daemon**. Allowed actions: `remove, quarantine, pass, report` Default value: **`SkipObject`** `= report`

Parameter	Description
`DaemonError = {action}`	Action applied to files which have caused errors during scanning by **Dr.Web Daemon**. Allowed actions: `remove, quarantine, pass, report` Default value: **`DaemonError`** `= quarantine`
`LicenseError = {action}`	Action applied to files which have not been scanned by **Dr.Web Daemon** due to license restrictions. Allowed actions: `remove, quarantine, pass, report` Default value: **`LicenseError`** `= report`
`ProcessingError = {action}`	Action applied to files which have not been processed by **Dr.Web NSS** due to errors. Allowed actions: `remove, quarantine, pass, report` Default value: **`ProcessingError`** `= report`

[Stat] Section

In this section you can specify settings for statistics gathering.

Parameter	Description
	`[Stat]`
`SendToAgent = {logical}`	Enables or disables sending statistics on **Dr.Web NSS** operation to **Dr.Web Agent**. If parameter value is set to `No`, statistics

Parameter	Description
	will not be gathered. Default value: **SendToAgent** = yes
SendPeriod = {time}	Time period for sending general statistics to **Dr.Web Agent**. Default value: **SendPeriod** = 5m

[Quarantine] Section

In this section you can specify setting for **Quarantine** used to isolate infected or suspicious files.

Parameter	Description
[Quarantine]	
Path = {path to directory}	Path to directory of the **Quarantine**. **Dr.Web NSS** module must have permissions for creating, change, delete and read files in this directory. Default value: **Path** = /var/drweb/infected/nss
FilesMode = {numerical value}	Permissions for files to be moved to the **Quarantine**. Default value: **FilesMode** = 0660

[Notifications] Section

In this section you can specify settings for notifications sent on various events (scanning and processing errors, detection of malware etc.).

Parameter	Description
`[Notifications]`	
`ExternalProgram = {String}`	Command for execution of external program then action (`remove`, `quarantine`, `cure`, `report`) has been applied to a file. Event which caused the action will be logged after execution of the program. A thread executing this command is waiting for it to finish, and if its return code is not zero, appropriate message will be written to the log file. You can use the following macros in this command: • **$HOSTMASTER$** - **Hostmaster**; parameter value; • **$REASON$** - name of the event; • **$ACTION$** - name of the action applied at this event; • **$VERSION$** - current version number; • **$FILE$** - full path to the file which caused the event; • **$SIZE$** - size (in bytes) of the file which caused the event; • **$TIME$** - local server time of command execution; • **$DAEMON_REPORT$** - **Dr.Web Daemon** report received after scanning the file. May be empty. Lines in report are delimited with line feeds; • **$VIRUSES$** - list of viruses found during scanning. May be empty. Viruses are delimited with commas.

Parameter	Description
	Example: (must be specified in one line): `"kdialog --passivepopup \"<html><font color=\"red\" size=\"5\">Attention, $REASON$ event is occured!</font> File <font color=\"blue\">$FILE$ (size=$SIZE$)</font> action=$ACTION$ </html>\" 10"` In KDE environment a pop-up window will appear whenever event (errors, detection of malware etc.) will occur. Default value: **ExternalProgram** =
SendMail = {logical}	Enables or disables sending of e-mail messages after applying remove, quarantine, cure or report actions. Command for sending e-mail notifications is executed after applying an action but before writing to the log file. E-mail notification will be sent to address specified in **Hostmaster** parameter. Templates for notifications are taken from directory specified in **Templates** parameter. Default value: **SendMail** = No
Templates = {path to directory}	Path to directory containing notificaion templates. Currently it should contain just **email.templ** template. You can use in that template macros listed in **ExternalProgram** parameter description.

Parameter	Description
	Default value: **Templates** = /etc/drweb/templates/nss
Hostmaster = {e-mail address}	E-mail address for sending e-mail notifications. Default value: **Hostmaster** = root@localhost
MailCommand = {String}	Shell command used to send e-mail notifications to administrator. Default value: **MailCommand** = "/usr/sbin/sendmail -i -bm -f drweb-nss -- %s"

Dr.Web Updater

You can use **Dr.Web Updater** to update automatically virus databases and content-specific black and white lists of Internet resources for the **Dr.Web for Novell Storage Services** solution. Updating module is implemented as a console script `update.pl` written in Perl, and you can find it in the directory containing **Dr.Web for Novell Storage Services** executable files.

Dr.Web Updater is require installed **Perl** 5.8.0 or later.

Dr.Web Updater settings are stored in `[Updater]` section of the `drweb32.ini` configuration file in `%etc_dir` directory. If you want to use alternative configuration file, specify the full path to it with command line parameter at start.

To run the script use the following command:

```
$ %bin_dir/update.pl [parameters]
```

Allowed parameters are listed in chapter Command Line Parameters.

Updating

To ensure reliable protection **Dr.Web for Novell Storage Services** solution requires regular updates of virus databases.

Dr.Web for Novell Storage Services virus databases are stored in files with *.vdb extension. Update servers of **Dr.Web Global Updating System** (**Dr.Web GUS**) may also store them in lzma-archives. When new viruses are discovered, small files (only several KBytes in size) with database segments describing these viruses are released to provide quick and effective countermeasures.

Updates are the same for all supported platforms. There are daily "hot" updates (`drwtoday.vdb`) and regular weekly updates (`drwXXXYY.vdb`), where `XXX` is antivirus engine version number, and YY is a sequential number, beginning from 00 (for example, the

first regular update for version 6.0 will be named `drw60000.vdb`).

"Hot" updates are released daily or even several times a day to provide effective protection against new viruses. These updates are installed over the old ones: i.e. previous `drwtoday.vdb` file will be overwritten. When new regular update is released, all records from `drwtoday.vdb` are copied to `drwXXYY.vdb`, and new empty `drwtoday.vdb` file is issued.

If you want to update virus databases manually, you must install all missing regular updates first, and then overwrite `drwtoday.vdb` file.

To add the update to the main virus databases, place the corresponding file to the directory for **Dr.Web for Novell Storage Services** executable files (`/var/drweb/bases/` by default) or to any other directory specified in the configuration file.

Signatures for virus-like malicious programs (adware, dialers, hacktools, etc.) are supplied in two additional files - `drwrisky.vdb` and `drwnasty.vdb` - with the structure similar to virus databases. These files are also updated regularly: `dwrXXYYY.vdb` and `dwnXXYYY.vdb` are for regular updates, and `dwrtoday.vdb` and `dwntoday.vdb` are for "hot" updates.

From time to time (as new antivirus techniques are developed), new versions of the antivirus package are released, containing the updated algorithms, implemented in the antivirus engine **Dr.Web Engine**. At the same time, all released updates are brought together, and the new package version is completed with the updated main virus databases with descriptions of all known viruses. Usually, when upgrading the package to the new version the databases remain portable: i.e. new bases can be linked up to the old **Dr.Web Engine**. Please note that this does not guarantee detection or curing of new viruses, as it requires upgrading of algorithms in the **Dr.Web Engine**.

With regular updating virus databases have the following structure:

- `drwebase.vdb` – general virus database, received with the new version of the package;

- `drwXXXYY.vdb` – regular weekly updates;
- `drwtoday.vdb` – "hot" updates released daily or several times a day;
- `drwnasty.vdb` – general database of other malware, received with the new version of the package;
- `dwnXXXYY.vdb` – regular weekly updates for other malware;
- `dwntoday.vdb` – "hot" updates for other malware;
- `drwrisky.vdb` – general database of riskware, received with the new version of the package;
- `dwrXXXYY.vdb` – regular weekly updates for riskware;
- `dwrtoday.vdb` – "hot" updates for riskware.

Virus databases can be automatically updated using **Dr.Web Updater** module (`%bin_dir/update.pl`). After installation user crontab file `/etc/cron.d/drweb-update` will be created to run **Updater** every 30 minutes to ensure regular updates and maximum protection. You can modify this file to change update period.

Cron Configuration

A special file with user settings will be created in the `/etc/cron.d/` directory during installation of the software complex. It will enable interaction between **cron** and **Dr.Web Updater**.

In the task created for **crond**, vixie cron syntax is used. If you use different **cron** daemon, such as **dcron**, it is necessary to manually create a task to automatically start the **Dr.Web Updater** module.

Please note that by default **cron** daemon launch **Dr.Web Updater** module every 0 and 30 minutes of every hour. This can cause increased load on the update servers of **Dr.Web GUS** and cause

update delays. To avoid such situation, it is recomended to change default values to arbitrary.

Command Line Parameters

- `--help` – used to show brief usage summary.
- `--ini` – used to specify usage of another (not default) configuration file. To use another configuration file, specify full path to it with `--ini` command line parameter. If the name of the configuration file is not specified, `%etc_dir/drweb32.ini` is used.

  Example:

  ```
  $ /opt/drweb/update.pl --ini=/path/to/conf_file
  ```

- `--what` – allows to temporarily override value of **`Section`** parameter on **Updater**'s launch. Parameter will take effect until next start of the script. Possible values: `Scanner` or `Daemon`.

  Example:

  ```
  $ /opt/drweb/update.pl --what=Scanner
  ```

- `--components` – used to view a list of all product components available for update.

  Example:

  ```
  $ /opt/drweb/update.pl --components
  ```

- You can also use command line parameter `--not-need-reload`:
  - without this parameter all daemons of **Dr.Web for Novell Storage Services** will be restarted after **`update.pl`** script finishes its work, if some components of **Dr.Web for Novell Storage Services** have been updated, removed or added;
  - if `--not-need-reload` parameter is specified without any value, after the completion of **`update.pl`** script work, any daemon of **Dr.Web for Novell Storage Services** won't be

restarted;

- if `not-need-restart` parameter specify names of the daemons, they will not be restarted after the completion of **`update.pl`** script work. Names of non-restarted daemons must are listed with comma separation, without white spaces, case insensitive.

Example:

```
$ /opt/drweb/update.pl --not-need-reload=drwebd
```

Blocking Updates for Selected Components

You can configure **Dr.Web Updater** to block updates for selected components of your solution **Dr.Web for Novell Storage Services** .

To view the list of available components, use `--components` command-line parameter:

Example:

```
# ./update.pl --components

Available Components:
    agent
    drweb          (frozen)
    icapd          (frozen)
    vaderetro_lib
```

If updates for any component are blocked, that component will be marked as frozen. Frozen components will not be updated when **Dr.Web Updater** is ran.

Blocking updates

To block updates for specific component use `--freeze=<components>` command-line parameter, where `<components>` is a comma-delimited list of names of components to be frozen.

Example:

```
# ./update.pl --freeze=drweb
Updates for component 'drweb' are frozen.
Run command './updater --unfreeze=drweb' to start
updates again.
```

Unblocking updates

To once again enable updates for a frozen component, use `--unfreeze=<components>` command-line parameter, where `<components>` is a comma-delimited list of names of components to be unfrozen.

Example:

```
# ./update.pl --unfreeze=drweb
Updates for component 'drweb' are no longer frozen.
```

Unfreezing will not update the component.

Restoring Components

When updating components of your **Dr.Web for Novell Storage Services** solution, back-up copies will be saved in **Dr.Web Updater** working directory. It enables you to restore any component to its previous state in case there are some problems with the update.

To restore component to a previous state, use `--restore=<components>` command-line parameter, where `<components>` is a comma delimited list of components to be restored.

Example:

```
# ./update.pl --restore=drweb

Restoring backup for component 'drweb'...
Updates for component 'drweb' are frozen.
Run command './updater --unfreeze=drweb' to start
updates again.

Backup for component 'drweb' has been restored!
Dr.Web (R) restore details:

Following files has been restored:
      /var/drweb/bases/drwtoday.vdb
      /var/drweb/bases/dwntoday.vdb
      /var/drweb/bases/dwrtoday.vdb
      /var/drweb/bases/timestamp
      /var/drweb/updates/timestamp
```

On restoring component will be automatically frozen. To enable updates for a restored component you need to unfreeze it.

Configuration File

Dr.Web Updater settings are stored in `Updater` section of configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory:

`Section [Updater]`

UpdatePluginsOnly = {logical}	With `Yes` value specified **Dr.Web Updater** will not update **Dr.Web Daemon** and **Dr.Web Scanner**. It will update only plug-ins. Default value: **UpdatePluginsOnly** = No

Section = {Daemon | Scanner}	Specifies from which section of configuration file **Dr.Web Updater** will take settings to determine path to key file, paths to virus databases, etc. Possible values: `Scanner, Daemon`. Value of this parameter can be temporarily overriden by `--what` command line parameter. Parameter will take effect until next start of the script. Default value: **Section** = Daemon
ProgramPath = {path to file}	Path to **Dr.Web Daemon** or **Dr.Web Scanner**. It is used by **Dr.Web Updater** for getting the product version and API information of the installed executable file. Default value: **ProgramPath** = %bin_dir/drwebd
SignedReader = {path to file}	Path to program which is used to read digitally signed files. Default value: **SignedReader** = %bin_dir/read_signed
LzmaDecoderPath = {path to directory}	Path to the directory containing program used for unpacking of lzma-archives. Default value: **LzmaDecoderPath** = %bin_dir/
LockFile = {path to file}	Path to lock file used to prevent sharing of certain files during their processing by **Dr.Web Updater**. Default value: **LockFile** = %var_dir/run/update.lock

CronSummary = {logical}	If you specify Yes, **Dr.Web Updater** will output update report for each session to stdout. This mode can be used to send notifications to administrator by email, if **Dr.Web Updater** is run by the cron daemon. Default value: **CronSummary** = Yes
DrlFile = {path to file}	Path to the file (*.drl) containing list of accessible updating servers of **Dr.Web GUS**. **Dr.Web Updater** randomly selects a server from this list to download updates. Detailed about updates downloading see in chapter Updating Process. This file is signed by **Doctor Web** and should not be modified by the user. It is updated automatically. Default value: **DrlFile** = %var_dir/bases/update.drl
CustomDrlFile = {path to file}	Path to the file (*.drl) with the alternative list of accessible updating servers of **Dr.Web GUS**. **Dr.Web Updater** also randomly selects a server from this list to download updates. Detailed about updates downloading see in chapter Updating Process. This file is signed by **Doctor Web** and should not be modified by the user. It is updated automatically. Default value: **CustomDrlFile** = %var_dir/bases/custom.drl

FallbackToDrl = {logical}	To allow using of file from **DrlFile** in case it was not possible to connect to no one of the servers which are listed in the **CustomDrlFile**. If specified value is No, file specified in **DrlFile** is not used. In case the file specified in **CustomDrlFile** doesn't exist, file specified in **DrlFile** is used regardless of value specified for **FallbackToDrl** parameter. Detailed about updates downloading see in chapter Updating Process. Default value: **FallbackToDrl** = Yes
DrlDir = {path to directory}	Path to the directory containing drl files with lists of update servers **Dr.Web GUS** for each plug-in. These files are signed by **Doctor Web** and should not be modified by the user. Default value: **DrlDir** = %var_dir/drl/
Timeout = {numerical value}	Maximum waiting period for downloading updates from selected server of **Dr.Web GUS** in seconds. Default value: **Timeout** = 90
Tries = {numerical value}	Number of attempts to be made by **Dr.Web Updater** to establish a connection with the selected update server. Default value: **Tries** = 3

ProxyServer = {host name | IP address}	Host name or IP address of the proxy server which is used for Internet access. If the proxy server is not used the value of this parameter must be empty. Default value: **ProxyServer** =
ProxyLogin = {string}	User login for used proxy server (if it requires authentication). Default value: **ProxyLogin** =
ProxyPassword = {string}	The password for used proxy server (if it requires authentication). Default value: **ProxyPassword** =
LogFileName = {syslog | file name}	Path to log file name. You can specify `syslog` as log file name and logging will be carried out by `syslogd` system service. In this case you must also specify **SyslogFacility** and **SyslogPriority** parameters. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label which is used by `syslogd` system service. Default value: **SyslogFacility** = Daemon
LogLevel = {log level}	Log verbosity level.

	Following levels are allowed: • Quiet • Error • Warning • Info • Debug • Verbose Default value: **LogLevel** = Info
BlacklistPath = {path to directory}	Path to directory with .dws files. Default value: **BlacklistPath** = %var_dir/dws
AgentConfPath = {path to file}	Path to **Dr.Web Agent** configuration file. Default value: **AgentConfPath** = %var_dir/agent.conf
ExpiredTimeLimit = {numerical value}	Number of days before license expiration during which **Dr.Web Updater** will be attempting to update license key file. Default value: **ExpiredTimeLimit** = 14
ESLockfile = {path to file}	Path to lock file. If the lock file exists, **Dr.Web Updater** will not be automatically initialized by cron daemon. Default value: **ESLockfile** = %var_dir/run/es_updater.lock

Updating Process

Updating is done in following stages:

1. **Dr.Web Updater** is reading configuration file (`drweb32.ini` by default, or specified with `--ini` command line argument).
2. **Dr.Web Updater** uses parameters from section `[Updater]` of configuration file (see description above), and parameters **EnginePath**, **VirusBase**, **UpdatePath** and **PidFile**.
3. **Dr.Web Updater** selects updates server of **Dr.Web GUS** for updates downloading. The updates server will be selected by following way:
   - Reading the files with lists of the updates servers (`drl`), specified in the **DrlFile** and **CustomDrlFile** parameters;
   - If both files are not accessible, then updating process will be stopped and ended;
   - If only one (any) file is accessible, then it will be used regardless of value specified for **FallbackToDrl** parameter;
   - If both files are accessible, then updates servers will be selected at beginning from file specified in **CustomDrlFile** parameter;
   - If it is not possible to connect to no one of the servers from the file specified in **CustomDrlFile** parameter, and value of **FallbackToDrl** is specified to `Yes`, then servers from the file specified in **DrlFile** parameter, will be tried to connecting. In opposite case updating will be stopped and ended.
4. **Dr.Web Updater** make connection attempts to random chosen servers from the selected list until connection attempt to the server won't appear successful (at connection attempt **Dr.Web Updater** waits the answer from the server during the time period specified in the **Timeout** parameter).

5. Module requests from connected server of **Dr.Web GUS** the list of available updates, and then lzma archives of its. In case archives are not presented on the server, the updates will be downloaded as `vdb` files. For lzma-archives unpacking **`lzma`** utility is used. Path to the directory which contains utility is specified in **`LzmaDecoderPath`** parameter.
6. Received (and unpacked) updates will be saved in directories as described in chapter Updating.

Dr.Web Monitor

Dr.Web Monitor component is presented by a memory resident module **drweb-monitor**.

It is used to increase fault-tolerance of the whole **Dr.Web for Novell Storage Services** suite. It ensures correct startup and termination of operation of software modules and their components as well as restart of any component due to its abnormal operation. **Dr.Web Monitor** starts all modules and loads, if necessary, some extra components of these modules. If **Dr.Web Monitor** fails to start a module, it repeats an attempt later. Number of attempts and a time period between them are defined by **Dr.Web Monitor** settings.

After all modules are loaded, **Dr.Web Monitor** permanently controls their operation. If any module or one of its components operates abnormally, **Dr.Web Monitor** restarts the stalled application. Maximum number of attempts to restart a component and a period of time between them are defined by **Dr.Web Monitor** settings. If any of the modules starts to operate abnormally, **Dr.Web Monitor** notifies the system administrator.

Dr.Web Monitor can interact with **Dr.Web Agent** by exchanging control signals.

Operation Mode

If necessary, **Doctor Web** solutions can be used to connect to corporate or private **Anti-virus networks** managed by **Dr.Web Enterprise Security Suite**. To operate in such central protection mode, you do not need to install additional software or uninstall your **Dr.Web** solution.

To provide you with this option, **Dr.Web Monitor** can operate in one of the two following modes:

- **Standalone mode** when a protected computer is not included in an anti-virus network or managed remotely. In this

mode, configuration files and key files reside on local drives, **Dr.Web Monitor** is controlled in full from the protected computer, and modules start as set in **Dr.Web Monitor** configuration file.

- **Enterprise mode** (or **central protection mode**) when protection of local computer is managed from a central protection server. In this mode, some features and settings of **Dr.Web for Novell Storage Services** may be modified and blocked for compliance with a general (e.g., company) security policy. A key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used.

To use central protection mode

1. Contact anti-virus network administrator of your company for a public key file and parameters of connection to the central protection server.
2. In **Dr.Web Monitor** configuration file (by default, `%etc_dir/monitor.conf`), set the **`UseEnterpriseMode`** parameter to `Yes`.

   In the central protection mode, some features and settings of **Dr.Web for Novell Storage Services** may be modified and blocked for compliance with the general security policy. A key file for operation in this mode is received from central protection server. Your personal key file on a local computer is not used.

For **Dr.Web for Novell Storage Services** to fully support central protection mode, you must also set **Dr.Web Agent** to operate in enterprise mode. For more details, see Operation Mode for **Dr.Web Agent**.

To use standalone mode

1. Make sure that all necessary modules that you want **Dr.Web Monitor** to start are listed in the **`RunAppList`** parameter under the `[Monitor]` section of **Dr.Web Monitor** configuration file (by default, `%etc_dir/monitor.conf`).

2. In the `[Monitor]` section of **Dr.Web Monitor** configuration file, set the **`UseEnterpriseMode`** parameter to `No`.

   On switching to this mode, all settings of **Dr.Web for Novell Storage Services** are unlocked and restored to their previous or default values. You can once again access all features of **Dr.Web for Novell Storage Services** solution and configure them in full.

For correct operation in standalone mode, **Dr.Web for Novell Storage Services** requires a valid personal key file. The key files received from central protection server cannot be used in this mode.

Command Line Parameters

To run **Dr.Web Monitor**, use the following command:

```
drweb-monitor [parameters]
```

where the following `parameters` are allowed:

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and terminate module		
`-v`	`--version`	
Description: Show **Dr.Web Monitor** version on the screen and terminate module		
`-u`	`--update`	
Description: Start updating all components of **Dr.Web for Novell Storage Services**		

Short case	Extended case	Arguments
-C	--check-only	
Description: Check correctness of **Dr.Web Monitor** configuration. This parameter can't be used if in the system already exist started copy of **Dr.Web Monitor** process		
-A	--check-all	<path to file>
Description: Check correctness of configurations of all components of **Dr.Web for Novell Storage Services**		
-c	--conf	<path to file>
Description: Module must use the specified configuration file		
-r	--run	<application name>[, <application name>,...]
Description: Run applications, name of which are specified. As an application name must be used string <application name> from Application "<application name>" section header in mmc file (about this see in chapter Interaction with other Software Modules). This parameter can't be used if in the system already exist started copy of **Dr.Web Monitor** process.		

Usage example:

```
drweb-monitor -r AGENT, NSS
```

Configuration File

Setup of **Dr.Web Monitor** is performed using its configuration file %etc_dir/monitor.conf.

General principles of the **Dr.Web for Novell Storage Services** configuration files organization see in chapter Configuration files.

[Logging] Section

In [**Logging**] section parameters responsible for logging information about operation of **Dr.Web Monitor** are collected:

[Logging]

Level = {log level}	**Dr.Web Monitor** log verbosity level. Following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **Level** = Info
IPCLevel = {log level}	Log verbosity level of IPC library. Following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **IPCLevel** = Error
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon
FileName = {syslog | path to file}	Path to log file name. You can specify syslog as log file name

	and logging will be carried out by `syslogd` system service. In this case you must also specify **`SyslogFacility`** parameter. Default value: **`FileName`** = `syslog`

[Monitor] Section

[`Monitor`] section contains all **Dr.Web Monitor** main settings:

[`Monitor`]

`RunForeground` = `{logical}`	**Yes** value forces **Dr.Web Monitor** not to use `daemon` mode. It helps to control its state using special utilities (i.e., **`daemontools`**). Default value: **`RunForeground`** = `No`
`User` = `{text value}`	User name used to run **Dr.Web Monitor** with certain user privileges. Default value: **`User`** = `drweb`
`Group` = `{text value}`	User group name used to run **Dr.Web Monitor** with certain user privileges. Default value: **`Group`** = `drweb`

PidFileDir = {path to directory}	Path to directory where PID-file of **drweb-monitor** is stored when **Dr.Web Monitor** is started. Default value: **PidFileDir** = %var_dir/run/
ChDir = {path to directory}	Change of working directory when **Dr.Web Monitor** is started. If this parameter is set up, **Dr.Web Monitor** changes directory to the one specified in this parameter value. Otherwise working directory is not changed. Default value: **ChDir** = /
MetaConfigDir = {path to directory}	Path to directory where meta-configuration files reside. These files contain settings defining **Dr.Web Monitor** interaction with other modules of **Dr.Web** suite. Meta-configuration files are supplied by **Dr.Web Dr.Web** developers and don't need editing. Default value: **MetaConfigDir** = %etc_dir/monitor/
Address = {address}	Socket used by **Dr.Web Monitor** to receive control signals from other components of **Dr.Web** suite. Default value: **Address** = local:%var_dir/ipc/.monitor
Timeout = {numerical value}	Maximum time in seconds to establish connection between **Dr.Web Monitor** and other components of **Dr.Web** suite.

	Default value: **Timeout** = 5
TmpFileFmt = {text value}	Template of names of **Dr.Web Monitor** temporary files. Template layout: path_to_file.XXXXXX where X – random symbol (letter or digit), used in temporary file names. Default value: **TmpFileFmt** = %var_dir/msgs/tmp/monitor.XXXXXX
RunAppList = {text value}	List of modules started by **Dr.Web Monitor**, with comma used as a delimiter. Please note that this parameter will not be modified after uninstalling **Dr.Web** modules. You must manually remove uninstalled modules from this parameter. Otherwise **Dr.Web Monitor** will no be able to run and to execute other **Dr.Web** modules. Default value: **RunAppList** = AGENT
UseEnterpriseMode = {logical}	Yes value makes **Dr.Web Monitor** receive the list of modules to be started from **Dr.Web Agent**, not from **RunAppList** parameter value. Default value: **UseEnterpriseMode** = No
RecoveryTimeList = {numerical values}	Time intervals between attempts to restart not responding modules (in seconds). This parameter can have multiple values, delimited by commas. First attempt to restart a module is made after a period of time specified in first parameter value, second attempt – using second parameter

	value, and so on. Default value: **RecoveryTimeList** = 0,30,60
InjectCmd = {string}	Command to send reports. Please note that if you want to send reports to some other address (not only to root@localhost), you should specify it in the command. Default value: **InjectCmd** = "/usr/sbin/sendmail -t"
AgentAddress = {address}	Socket used by **Dr.Web Monitor** to interact with **Dr.Web Agent** (parameter value must be the same as **Address** parameter value from **Dr.Web Agent** configuration file). Default value: **AgentAddress** = local:%var_dir/ipc/.agent
AgentResponseTime = {numerical value}	Maximum time to get a response from drweb-agent module in seconds. If **Dr.Web Agent** doesn't respond during this period of time, **Dr.Web Monitor** considers drweb-agent not working and tries to restart it. If 0 is specified, response time is unlimited. Default value: **AgentResponseTime** = 5

Running Dr.Web Monitor

When **Dr.Web Monitor** is ran with default settings the following actions are performed:

1. **Dr.Web Monitor** searches for and loads its configuration file. If configuration file is not found, loading stops;
2. Then it enters `daemon` mode, so all information about loading problems cannot be output to console anymore and is written to log file;
3. Socket for **Dr.Web Monitor** interaction with other software modules is created. If TCP socket is used, there can be several connections (loading continues if at least one connection is established). If UNIX socket is used, it can be created only if the user whose privileges are used to run **drweb-monitor** has read and write access to the certain directory. If socket cannot be created, loading stops;
4. PID-file with **drweb-monitor** PID information is created. If PID-file cannot be created, loading stops;
5. **drweb-monitor** module starts other software modules. If some module cannot load, **Dr.Web Monitor** tries to restart it. If all **Dr.Web Monitor** attempts to start the module are unsuccessful, **Dr.Web Monitor** unloads all previously loaded modules and terminates. All problems with the startup of modules **Dr.Web Monitor** reports using one of available methods (output to log file, notification via e-mail, startup of specific program). Notification methods used for various modules are set in **Dr.Web Monitor** meta-configuration file.

For successful startup of **Dr.Web Monitor** in automatic mode:

- value of `ENABLE` variable must be changed to `1` in the **drweb-monitor** `enable` file .

Please note that if you select "Configure Services" option in the conversation with the post-install script, all services including **Dr.Web Agent** will be started automatically.

Placement of the enable files is depends from **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:

  Files will be placed to directory `%etc_dir` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

- Installation from **native DEB packages**:

  Files will be placed to directory `/etc/defaults` and will be named
  `drwebd`,
  `drweb-monitor`.

- Installation from **native RPM packages**:

  Files will be placed to directory `/etc/sysconfig` and will be named
  `drwebd.enable`,
  `drweb-monitor.enable`.

Interaction with other Software Modules

Interaction with other software modules is performed via *Monitor configuration files* (`mmc`-files). These files are included in packages of those products which can interact with **Dr.Web Monitor**. In these files components' contents, location of binaries, their starting sequence and parameters of startup are described.

Description of each component can be found in `Application` section named after this component. At the end of the section `EndApplication` must be specified.

The following parameters must be present in the description of the component:

- **FullName** – full name of the component.
- **Path** – path to binary files.
- **Depends** – names of components which must be started before the described component is started. For example, AGENT component must be started before **Dr.Web Daemon** component, therefore in mmc-file for **Dr.Web Daemon** **Depends** parameter has AGENT value. If there are no dependencies, this parameter can be skipped.
- **Components** – list of binary files of modules started when component itself is started. Modules are started in order they are specified in this parameter. For each module command line parameters (may be enclosed in quotation marks), timeouts for startup and close down, notification type and startup privileges. *Notification type* – defines where to send notifications about component's failure. When MAIL value is specified, notifications are sent by mail, when LOG value is specified, information is output to log only. *Startup privileges* – define with privileges of which group and user, the component will be started.

Example of mmc-file for Dr.Web Daemon:

```
Application "DAEMON"
    FullName    "Dr.Web (R) Daemon"
    Path        "/opt/drweb/"
    Depends     "AGENT"
    Components
          # name     args     MaxStartTime
 MaxStopTime     NotifyType  User:Group
          drwebd   "-a=local:/var/drweb/ipc/.
agent --foreground=yes"   30  10  MAIL  drweb:
drweb
    EndComponents
EndApplication
```

Dr.Web Agent

Dr.Web Agent is a resident module used to manage settings of various modules of **Dr.Web for Novell Storage Services** solution, define antivirus policy depending on available licenses and collect virus statistics. When separate modules of **Dr.Web for Novell Storage Services** are started, or settings are changed, **Dr.Web Agent** sends to these modules all necessary configuration information. **Dr.Web Agent** can interact with other modules by exchanging control signals.

Since all the components of **Dr.Web for Novell Storage Services** solution (except for **Dr.Web Monitor**) receive their settings via **drweb-agent** module, it must be ran before all these modules, but after the **drweb-monitor** module.

Pleas note, that when several parameters with the same name are specified in configuration file, **Dr.Web Agent** unites them in one string with comma as delimiter. You can also use backslash symbol "\" to define parameter value in several lines. New line after backslash will be added to the previous line when **Dr.Web Agent** reads configuration information.

Dr.Web Agent can operate in two modes: `Standalone` and `Enterprise`. When **Dr.Web for Novell Storage Services** solution is integrated with **Dr.Web Enterprise Security Suite** (**Dr.Web ESS** hereinafter), **Dr.Web Agent** works in `Enterprise` mode. In this mode **Dr.Web Agent** connects to **Dr.Web Enterprise Server** (**Enterprise Server** hereinafter) server and receives licence key files and settings of antivirus modules. **Dr.Web Agent** itself is controlled via **Dr.Web ESS** server from antivirus console. To run **Dr.Web Agent** in `Enterprise` mode, you must set up correctly all the parameters from `[EnterpriseMode]` section of the **Dr.Web Agent**'s configuration file.

If **Dr.Web for Novell Storage Services** solution is used on a workstation not included in **Dr.Web ESS** antivirus network, then **Dr.Web Agent** works in `Standalone` mode. In this mode configuration files and license key files reside on local drives, and **Dr.Web Agent** can be controlled directly from this computer.

Presently **Dr.Web for Novell Storage Services** does not support integration with **Dr.Web Enterprise Security Suite**. **Dr.Web Agent** can be ran in standalone mode only.

Operation Mode

If necessary, **Doctor Web** solutions can be used to connect to corporate or private anti-virus networks managed by **Dr.Web Enterprise Security Suite** (**Dr.Web ESS**). To operate in such central protection mode, you do not need to install additional software or uninstall your **Dr.Web** solution.

To provide you with this option, **Dr.Web Agent** can operate in one of the two following modes:

- **Standalone mode** when a protected computer is not included in an anti-virus network or managed remotely. In this mode, configuration files and key files reside on local drives, and **Agent** is controlled in full from the protected computer.
- **Enterprise mode** (or central protection mode), when protection of the computer is managed from a central protection server. In this mode, some features and settings of **Dr.Web for Novell Storage Services** may be modified and blocked for compliance with a general (e.g., company) security policy. Licence key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used

To use central protection mode

1. Contact the anti-virus network administrator of your company for a public key file and parameters of connection to the central protection server.
2. In the **Dr.Web Agent** configuration file (by default, `%etc_dir/agent.conf`), set up the following parameters in the `[EnterpriseMode]` section:

- Set the **PublicKeyFile** parameter to location of a public key file received from anti-virus network administrator (usually, %var_dir/drwcsd.pub).This file includes an encryption public key for the access to **Dr.Web ESS**. If you are the anti-virus network administrator, you can locate the file in the corresponding directory on the **Enterprise Server**.
- Set the **ServerHost** parameter to IP-address or host name of the **Enterprise Server**.
- Set the **ServerPort** parameter to the **Enterprise Server** port number (usually, 2193).

3. To connect to central protection server, set the **UserEnterpriseMode** parameter to Yes.

   In the central protection mode, some features and settings of **Dr.Web for Novell Storage Services** may be modified and blocked in compliance with the general security policy. A key file for operation in this mode is received from central protection server. Your personal key file on a local computer is not used.

To run **Dr.Web Agent** in central protection mode drweb-agent-es package must be installed.

For **Dr.Web for Novell Storage Services** to fully support central protection mode, you must also set **Dr.Web Monitor** to operate in enterprise mode. For more details, see Operation Mode for **Dr.Web Monitor**.

To use standalone mode

1. Make sure that all parameters in the [StandaloneMode] section of the **Dr.Web Agent** configuration file (by default, %etc_dir/agent.conf) are set properly.
2. In the [EnterpriseMode] section of the **Dr.Web Agent** configuration file, set the **UseEnterpriseMode** parameter to No.

   On switching to this mode, all settings of **Dr.Web for Novell**

Storage Services are unlocked and restored to their previous or default values. You can once again access all features of **Dr.Web for Novell Storage Services** solutions and configure them in full.

For correct operation in standalone mode, **Dr.Web for Novell Storage Services** requires a valid personal key file. The key files received from central protection server cannot be used in this mode.

Joint usiage of Dr.Web for Novell Storage Services and Dr.Web Anti-virus for Linux solutions in central protection mode

You can safely use all server UNIX **Dr.Web** solution on the single host in central protection mode. **Dr.Web Anti-virus For Linux** however will be in conflict with server solutions. To run **Dr.Web for Novell Storage Services** or other **Dr.Web** server solutions in the central protection mode on the same host with **Dr.Web Anti-virus For Linux**, you will need to rename amc files for **Dr.Web Anti-virus For Linux** (`drweb-cc.amc`, `drweb-spider.amc`).

Due to implementation details, it is not possible to run **Dr.Web for Novell Storage Services** and **Dr.Web Anti-virus for Linux** in central protection mode on one host simultaneously. To enable central protection mode in **Dr.Web for Novell Storage Services** you should turn **Dr.Web Anti-virus for Linux** to standalone mode and delete or move to another directory files `%etc_dir/agent/drweb-cc.amc` and `%etc_dir/agent/drweb-spider.amc`.

It is recommended to keep this files as a backup in directory other than `%etc_dir/agent` if you are going to run **Dr.Web Anti-virus for Linux** in central protection mode later. In this case, set **Dr.Web for Novell Storage Services** into standalone mode, copy back-ups of `drweb-cc.amc` and `drweb-spider.amc` files to directory `%etc_dir/agent/` and follow the instructions given in **Dr.Web Anti-virus for Linux** User Guide.

Command Line Parameters

To run **Dr.Web Agent**, use the following command:

```
drweb-agent [parameters]
```

where the following `parameters` are allowed:

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and terminate module		
`-v`	`--version`	
Description: Show **Dr.Web Agent** version on the screen and terminate module		
`-u`	`--update-all`	
Description: Start updating all components of **Dr.Web for Novell Storage Services**		
`-f`	`--update-failed`	
Description: Start updating all components of **Dr.Web for Novell Storage Services**, for which updating in standard mode was failed		
`-C`	`--check-only`	
Description: Check correctness of **Dr.Web Agent** configuration. This parameter can't be used if in the system already exist started copy of **Dr.Web Agent** process		
`-c`	`--conf`	`<path to file>`
Description: Module must use the specified configuration file		
`-d`	`--droppwd`	

Short case	Extended case	Arguments
Description: Discard registration data used for connecting to **Dr.Web Enterprise Server** (username, password). At the next attempt of connecting to **Dr.Web Enterprise Server** will be started process of new workstation registration		
-p	--newpwd	
Description: Change username and password for connecting to **Dr.Web Enterprise Server**		
-s	--socket	<path to file>
Description: Use for interaction with controlled modules the specified socket		
-P	--pid-file	<path to file>
Description: Use the specified file as PID file of **Dr.Web Agent**		
-e	--export-config	<application name>
Description: Export configuration of the application, name of which is specified, to **Dr.Web Enterprise Server**. As an application name must be used string <application name> from Application "<application name>" section header in amc file (about this see in chapter Interaction with other Software Modules). This parameter can't be used if in the system already exist started copy of **Dr.Web Agent** process. Also it can't be used for export of **Dr.Web Antivirus for Linux** configuration.		

Configuration File

Setup of **Dr.Web Agent** is performed using its configuration file %etc_dir/agent.conf.

General principles of the **Dr.Web for Novell Storage Services** configuration files organization see in chapter Configuration files.

[Logging] Section

In [**Logging**] section parameters responsible for logging information about operation of **Dr.Web Agent** are collected:

[Logging]

Level = {log level}	**Dr.Web Agent** log verbosity level. Following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **Level** = Info
IPCLevel = {log level}	Log verbosity level of IPC library. Following levels are allowed: • Quiet • Error • Alert • Info • Debug Default value: **IPCLevel** = Error
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon
FileName = {path to file | syslog}	Path to log file name. You can specify syslog as log file name

	and logging will be carried out by `syslogd` system service. In this case you must also specify **`SyslogFacility`** parameter. Default value: **`FileName`** = `syslog`

[Agent] Section

This section contains general settings of **Dr.Web Agent**:

`[Agent]`

`MetaConfigDir` = `{path to directory}`	Directory name where meta-configuration files of **`drweb-agent`** reside. These files contain settings defining **Dr.Web Agent** interaction with other modules of **Dr.Web** software complex. Meta-configuration files are supplied by **Dr.Web** developers and do not need to be modified. Default value: **`MetaConfigDir`** = `%etc_dir/agent/`
`UseMonitor` = `{logical}`	`Yes` value tells **`drweb-agent`** that **Dr.Web Monitor** is used as a part of **Dr.Web for Novell Storage Services** solution. Default value: **`UseMonitor`** = `Yes`
`MonitorAddress` = `{address}`	Socket used by **Dr.Web Agent** to interact with **Dr.Web Monitor** (parameter value must be the same as **`Address`** parameter value from **Dr.Web Monitor** configuration file). Default value: **`MonitorAddress`** = `local:%var_dir/ipc/.monitor`

`MonitorResponseTime = {numerical value}`	Maximum time to get a response from **drweb-monitor** module in seconds. If **Dr.Web Monitor** doesn't respond during this period of time, **Dr.Web Agent** considers **drweb-monitor** not running and stops trying to establish connection with **Dr.Web Monitor**. Default value: **MonitorResponseTime** = 5
PidFile = `{path to file}`	Filename where **Dr.Web Agent** PID is written when **Dr.Web Agent** is run. Default value: **PidFile** = %var_dir/run/drweb-agent.pid

[Server] Section

In [**Server**] section parameters defining interaction of **Dr.Web Agent** with other modules of **Dr.Web for Novell Storage Services** solution are collected:

```
[Server]
```

Address = `{address}`	Socket used by **Dr.Web Agent** to interact with other modules of software complex. Multiple sockets can be specified, with comma used as a delimiter. Default value: **Address** = local:%var_dir/ipc/.agent, inet:4040@127.0.0.1
Threads = `{numerical value}`	Number of **drweb-agent** simultaneous threads. This parameter controls maximum number of simultaneous connections to modules reporting virus statistics to **Dr.Web Agent**.

	Value of this parameter cannot be changed using `SIGHUP` signal. If 0 is specified, number of threads is unlimited (not recommended). Default value: **`Threads`** = 2
`Timeout` = {numerical value}	Maximum time in seconds for establishing connection between **Dr.Web Agent** and other **Dr.Web** modules. If 0 is specified, time for establishing connection is unlimited. Default value: **`Timeout`** = 15

[EnterpriseMode] Section

[**EnterpriseMode**] section contains parameters defining **Agent** operation in **Enterprise** mode:

```
[EnterpriseMode]
```

UseEnterpriseMode = {logical}	With `Yes` value specified **Dr.Web Agent** works in `Enterprise` mode, with `No` value specified it works in `Standalone` mode. Default value: **`UseEnterpriseMode`** = No
`ComputerName` = {text value}	Computer name in **Anti-virus network**. Default value: **`ComputerName`** =

VirusbaseDir = {path to directory}	Path to directory where virus databases are located. Default value: **VirusbaseDir** = %var_dir/bases
PublicKeyFile = {path to file}	Path to file with public key to access **Dr.Web Enterprise Server**. Default value: **PublicKeyFile** = %bin_dir/drwcsd.pub
ServerHost = {IP address}	**Dr.Web Enterprise Server** IP address. Default value: **ServerHost** = 127.0.0.1
ServerPort = {port number}	Port number to access **Dr.Web Enterprise Server**. Default value: **ServerPort** = 2193
CryptTraffic = {Yes | Possible | No}	Encryption of traffic between **Dr.Web Enterprise Server** and **Dr.Web Agent**: • Yes – mandatory encryption • Possible – encryption if it is possible • No – do not encryption Default value: **CryptTraffic** = possible
CompressTraffic = {Yes | Possible | No}	Compression of traffic between **Dr.Web Enterprise Server** and **Dr.Web Agent**: • Yes – mandatory compression • Possible – compression if it is possible • No – do not compression

	Default value: **CompressTraffic** = possible
CacheDir = {path to directory}	Path to directory, where different utility files are stored: configuration files, files with access privileges for applications managed by **Dr.Web Enterprise Server**, files with registration information on **Dr.Web Enterprise Server**, etc. Default value: **CacheDir** = %var_dir/agent

[StandaloneMode] Section

In [**StandaloneMode**] section parameters defining **Dr.Web Agent** operation in **Standalone** mode are collected:

[StandaloneMode]

StatisticsServer = {text value}	Address (URL) of virus statistics server If not specified, then statistics will not be sent. Default value: **StatisticsServer** = stat.drweb.com:80/update
StatisticsUpdatePeriod = {numerical value}	Period in minutes of statistics updating. Value cannot be great than 5 Default value: **StatisticsUpdatePeriod** = 10
StatisticsProxy = {hostname | IP address}	IP address or host name of proxy server for virus statistics sending. Please note that if the value is not explicitly specified, value of http_proxy environment variable is used. **Example**:

	`StatisticsProxy = localhost:3128` Default value: **`StatisticsProxy =`**
`StatisticsProxyAuth` `= {text value}`	Authentication string (`<username>:<password>`) for access to proxy server. **Example**: `StatisticsProxyAuth = test:testpwd` Default value: **`StatisticsProxyAuth =`**
`UUID` `= {text value}`	Unique user identifier for virus statistics server http://stat.drweb.com/. Please note that this parameter is mandatory for statistics transfer – so if you want to enable this function, you must specify personal `UUID` as a value of this parameter (`md5` sum of license key file is usually used for this purpose). Default value: **`UUID =`**
`LicenseFile` `= {paths to files}`	Location of **Dr.Web** license key files or demo key files. Paths in the list are separated by commas (if more than one). Default value: **`LicenseFile`** `= %bin_dir/drweb32.key`

[Update] Section

`[Update]` section contains parameters that define how to

perform update of **Dr.Web for Novell Storage Services** components via **Dr.Web Enterprise Server**:

`[Update]`

CacheDir = {path to directory}	Directory where **Dr.Web Agent** temporarily stores downloaded update files. Default value: **CacheDir** = %var_dir/updates/cache
Timeout = {numerical value}	Maximum time on seconds for **Dr.Web Agent** to process downloaded update files. If 0 is specified, time for process is unlimited. Default value: **Timeout** = 120
RootDir = {path to directory}	Path to root directory. Default value: **RootDir** = /

Refer to *Administrator Manual* for **Dr.Web ESS** for more information.

Running Dr.Web Agent

Please note that if you select "Configure Services" option in the conversation with the post-install script, all services including **Dr.Web Agent** will be started automatically.

When **Dr.Web Agent** starts with default settings, the following actions are performed:

- **Dr.Web Agent** searches and loads its configuration file. If the configuration file is not found, **Dr.Web Agent** terminates.
- If the parameters in the `[EnterpriseMode]` section are

set correctly and **Dr.Web for Novell Storage Services** solution is operating within **Anti-virus network**, then **Dr.Web Agent** starts in enterprise mode. Otherwise, if parameters in the `[Standalone]` section are set correctly, **Dr.Web Agent** starts in the standalone mode. If the parameters in the `[Standalone]` section are not set, **Dr.Web Agent** terminates.

- Socket for interaction of **Dr.Web Agent** with other **Dr.Web** modules is created. If a TCP socket is used, then there can be several connections (loading continues if at least one connection is established). If a UNIX socket is used, it can only be created if the user, whose privileges are used to run **`drweb-agent`**, has read and write access to its directory. If socket cannot be created, **Dr.Web Agent** terminates.

Further loading process depends on the selected operation mode.

If **Dr.Web Agent** operates in **enterprise mode**:

- **Dr.Web Agent** connects to **Dr.Web Enterprise Server**. If the server is unavailable or authorization process fails during first time connection, **Dr.Web Agent** terminates. If **Dr.Web Agent** had worked previously with this server, but it's temporary unavailable (for example, in the event of connection problems), **Dr.Web Agent** use backup copies of configuration files received from the server earlier.
- If connection is established, **Dr.Web Agent** receives key files and settings from **Dr.Web Enterprise Server**. After all setting and key files are received, **Dr.Web Agent** is ready for work.

If **Dr.Web Agent** operates in the **standalone mode**, then meta-configuration files that define **Dr.Web Agent** interaction with other **Dr.Web** modules are loaded. Location of meta-configuration files is set in the **`MetaConfigDir`** parameter in the `[Agent]` section of the **Dr.Web Agent** configuration file. When meta-configuration files are successfully loaded, **Dr.Web Agent** is ready for work.

Interaction with other Software Modules

Interaction with other software modules is performed by **Dr.Web Agent**'s metaconfiguration files (`amc`-files). These files describe configuration parameters, which values will be received by respective **Dr.Web** modules from **Dr.Web Agent**.

Description of each module can be found in `Application` section named after this module. At the end of the section **`EndApplication`** must be specified.

The following parameters must be present in the description of the module:

- **`id`**: identifier of the module in **Dr.Web ESS**.
- **`ConfFile`**: path to the configuration file of the module.
- **`Components`**: description of the component. At the end of this section `EndComponents` must be specified. For each component its name, list of the sections in the configuration file and the parameters in these sections necessary for proper operation of the component are specified. The list of sections and parameters is comma separated.
  To describe individual parameters properly you must specify full path to them (e.g. `Quarantine/Path`). In description of sections only their names must be specified (e.g. `General`).
  Back slash (\) in descriptions of sections and parameters is used to denote line breaks.
  If all settings from the configuration file are necessary to a component, it is enough to specify instead of the list of sections and/or parameters a path "/*".

Example of amc-file for Dr.Web NSS:

```
Application "NSS"
   id                108
   ConfFile          "/etc/drweb/drweb-nss.
conf"
```

```
   Components
      drweb-nss     General, Logging,
DaemonCommunication, NSS, Actions,\
                    Quarantine, Stat,
Notifications
    EndComponents
EndApplication
```

Integration with Dr.Web Enterprise Security Suite

There are two possible situations which require integration of **Dr.Web for Novell Storage Services** solution with **Dr.Web Enterprise Security Suite** from system administrator:

- Setup and initial configuration of **Dr.Web for Novell Storage Services** in existing **Dr.Web ESS** environment;
- Embedding of successfully functioning UNIX server with already installed and configured **Dr.Web for Novell Storage Services** solution in **Dr.Web ESS** environment.

To make **Dr.Web for Novell Storage Services** solution work in **Dr.Web ESS** environment, set up **Dr.Web Agent** and **Dr.Web Monitor** components for operation in `Enterprise` mode, and register the whole solution on **Dr.Web Enterprise Server**.

According to connection policy for new working stations (for more details refer to administrator manual for **Dr.Web Enterprise Security Suite)**, UNIX server can be connected to **Dr.Web Enterprise Server** in two different ways:

- when new account is created by central protection server automatically;
- when corresponding account is created by administrator manually.

Setup of Components

To start up in Enterprise mode after installation it is necessary to specify the changes in local configuration files of **Dr.Web Agent** and **Dr.Web Monitor**.

For Dr.Web Agent

In [EnterpriseMode] section of **Dr.Web Agent** configuration file %etc_dir/agent.conf set the following parameter values:

- **UseEnterpriseMode** = Yes;
- **PublicKeyFile** = %var_dir/drwcsd.pub (encryption public key for the access to central protection server. Take this file from the corresponding directory of **Dr.Web Enterprise Server** and move it to the specified path);
- **ServerHost** = IP-address or host name of **Dr.Web Enterprise Server**;
- **ServerPort** = **Dr.Web Enterprise Server** port (2193 by default).

For Dr.Web Monitor

In [Monitor] section of the **Dr.Web Monitor** configuration file %etc_dir/monitor.conf set the following parameter values:

- **UseEnterpriseMode** = Yes.

Automatic Creation of New Account by ES Server

When new account is created automatically:

1. When **Dr.Web Agent** is first started in Enterprise mode, it sends a request for the account details (station ID and password) to **Dr.Web Enterprise Server**;
2. If **Dr.Web Enterprise Server** is set to **Approve access manually** mode (used by default, for more details refer to administrator manual for **Dr.Web ESS**), system administrator

must confirm registration of new station via web interface **Dr.Web Control Center** during one minute from an emergence of corresponding request;

3. After first start **Dr.Web Agent** records hash of station ID and password into file named `pwd`. This file is created in the directory that specified in **`CacheDir`** parameter of the [`EnterpriseMode`] section (default value is `%var_dir/agent/`);
4. Data from this file is used every time **Dr.Web for Novell Storage Services** solution connects to **Dr.Web Enterprise Server**;
5. If you delete password file, repeated registration request will be made to **Dr.Web Enterprise Server** after next start of the **Dr.Web Agent**.

Manual Creation of New Account by Administrator

When new account is created manually:

1. Create new account on **Dr.Web Enterprise Server**: station ID is generated automatically and password must be specified manually (for more details refer to administrator manual for **Dr.Web ESS**).
2. Start **Dr.WebAgent** using command line parameter **`--newpwd`** (or **`-p`**) and type in the station ID and password. **Dr.Web Agent** records hash of station ID and password into file named `pwd`. This file is created in the directory that specified in **`CacheDir`** parameter of the [`EnterpriseMode`] section (default value is `%var_dir/agent/`).
3. Data from this file is used every time **Dr.Web for Novell Storage Services** solution connects to **Dr.Web Enterprise Server**.
4. If you delete password file, the registration must be performed once again (with next start of **Dr.Web Agent**).

Configuring Components via Dr.Web Control Center (embedded in Enterprise Security Suite)

Configuration of **Dr.Web for Novell Storage Services** and **Dr.Web Daemon** (antivirus plug-in, included in standard installation package) can be performed via **Dr.Web Control Center**.

In **Dr.Web Enterprise Security Suite** standard installation package the basic configuration files for **Dr.Web for Novell Storage Services** and **Dr.Web Daemon** components for **Linux**, **FreeBSD** and **Solaris** are included. When you configure certain components via web interface (**Dr.Web Control Center**), values of corresponding parameters are changed in these configuration files on **Dr.Web Enterprise Server**. After that every time the components start, **Dr.Web Agent** requests and receives configuration from **Dr.Web Enterprise Server**.

Export of Existing Configuration to ES Server

Automatic export of configuration settings from local computer to **Dr.Web Enterprise Server** is possible via **Dr.Web Agent** operating in `Enterprise` mode. To export configuration use command line parameter **`--export-config`** (or **`-e`**).

You must specify the name of the component (`DAEMON`, `NSS`).

Example:

```
# %bin_dir/drweb-agent --export-config NSS
```

Starting up the System

To start up the system:

1. In **Dr.Web Control Center** interface open the page with **Monitor** settings and check **Daemon** and **NSS** boxes to enable configuration of the corresponding components;
2. Start **Dr.Web Monitor** on local computer:

```
# /usr/local/etc/rc.d/00.drweb-monitor.sh start
```

Collection of Virus Statistics

Dr.Web Agent receives statistics on computer threats from controlled modules and sends it to the official **Doctor Web** website devoted to statistics: http://stat.drweb.com/ (if Internet connection is available) or to **Dr.Web ESS** (if **Dr.Web Agent** is operating in enterprise mode). **Dr.Web Agent** needs the *unique user identifier* (UUID) to connect to this website. By default, license key file MD5 sum is used as a UUID. Also you can get a personal UUID from **Doctor Web Technical Support**. In this case, your UUID must be specified explicitly in the **Agent** configuration file.

Statistics are gathered only for **Dr.Web** modules that receive settings from **Dr.Web Agent**. Information on how to set up interaction with **Dr.Web Agent** can be found in chapters describing these modules.

On the statistics website, you can find the aggregate statistics for computer threats for a given server or for all servers supported by **Dr.Web Anti-virus for UNIX** or by **Dr.Web for Novell Storage Services** solution with anti-virus plug-in. **Dr.Web Agent** can simultaneously process statistics for computer threats from several different **Dr.Web** products which are able to interact with **Dr.Web Agent**.

Statistics processing results contain information on the most frequently detected threats (overall percentage only for aggregate

statistics and also number of detections for personalized statistics) for a given period.

Statistics is available in both HTML and XML formal. The second option is especially convenient when this data is going to be published on another web site, since it can be transformed according to web site's concept and design.

To get aggregate statistics on computer threats for all supported servers, visit http://stat.drweb.com/. You can view a list of detected threats for all supported servers (in descending order) with overall percentage of detections.

This web page may render differently depending on used browser.

The following illustration shows threats statistics.

Figure 15. Computer threats statistics

To alter search parameters and to repeat search

1. Select either **Mail** or **Files** flags to get the statistics about the computer threats detected in the e-mails or in files.
2. In the drop-down lists for **Start date** and **End date**, select choose **start/end date** and **time** for the period of interest.
3. In the **Top** field, enter the required number of rows in the statistics table (most frequently the detected threats will be shown).
4. Select **Plot graph** if you want to view statistics in graphical form.
5. Click **Query**. The file with aggregate statistics in the XML form can be found at http://info.drweb.com/export/xml/top

Example:

```
<drwebvirustop period="24" top="5"
  vdbaseurl="http://info.drweb.com/
virus_description/"
  updatedutc="2009-06-09 09:32:02">
 <item>
  <vname>Win32.HLLM.Netsky</vname>
  <dwvlid>62083</dwvlid>
  <place>1</place>
  <percents>34.201062139103</percents>
 </item>
 <item>
  <vname>Win32.HLLM.MyDoom</vname>
  <dwvlid>9353</dwvlid>
  <place>2</place>
  <percents>25.1303270912579</percents>
 </item>
 <item>
  <vname>Win32.HLLM.Beagle</vname>
  <dwvlid>26997</dwvlid>
  <place>3</place>
  <percents>13.4593034783378</percents>
 </item>
 <item>
  <vname>Trojan.Botnetlog.9</vname>
  <dwvlid>438003</dwvlid>
  <place>4</place>
  <percents>7.8644592583328</percents>
 </item>
 <item>
  <vname>Trojan.DownLoad.36339</vname>
  <dwvlid>435637</dwvlid>
  <place>5</place>
  <percents>7.31494163115527</percents>
 </item>
</drwebvirustop>
```

In this file, the following XML attributes are used:

- `period` – duration (in hours) of the statistics collection process;
- `top` – number of the most frequently detected threats in the statistics table (number of rows);
- `updatedutc` – last statistics' update time;
- `vname` – threat name;
- `place` – virus place in the statistics;
- `percents` – percentage of the detections.

The value of the period parameter and the sample size cannot be changed by user.

To get personalized threat statistics

Visit one of the following Web pages:

- For the statistics in HTML, go to http://stat.drweb.com/view/<UUID>. Personalized threat statistics page is similar to the aggregate threat statistics page.
- For the file with the personalized threat statistics in XML form, go to http://stat.drweb.com/xml/<UUID>.

The <UUID> in both cases stands for the MD5 sum of your license key file (unless you have a personal UUID received from **Doctor Web Technical Support**).

Example:

```
<drwebvirustop period="24" top="2"
user="<UUID>"
  lastdata="2005-04-12 07:00:00+04">
 <item>
  <caught>69</caught>
  <percents>24.1258741258741</percents>
  <place>1</place>
  <vname>Win32.HLLM.Netsky.35328</vname>
 </item>
 <item>
  <caught>57</caught>
  <percents>19.9300699300699</percents>
  <place>2</place>
  <vname>Win32.HLLM.MyDoom.54464</vname>
 </item>
</drwebvirustop>
```

In this file the following XML attributes are used:

- `period` – duration (in hours) of the statistics collection process;
- `top` – number of the most frequently detected threats in the statistics table (number of rows);
- `user` – user identifier;
- `lastdata` – last time user sent the data to the server;
- `vname` – threat name;
- `place` – threat place in the statistics;
- `caught` – a number of the detections of the certain threat;
- `percents` – percentage of the detections.

The value of the period parameter and the sample size cannot be changed by user.

Dr.Web Daemon

Dr.Web Daemon is a background antivirus module **drwebd**, designed to perform scanning for viruses on request from other **Dr.Web** software components. It can scan files on disk or data transferred through socket. Requests for scanning are sent using special protocol via UNIX sockets or TCP sockets. **Dr.Web Daemon** uses the same antivirus engine (**Dr.Web Engine**) and virus databases as **Dr.Web Scanner** and is able to detect and cure all known viruses.

Dr.Web Daemon is always running and has simple and straightforward protocol for sending scanning requests. Because of that, it is a perfect solution to be used as antivirus filter for

Command-line Parameters

To run **Dr.Web Daemon**, use the following command:

```
drwebd [parameters]
```

where the following `parameters` are allowed:

Short case	Extended case	Arguments
`-h, -?`	`-help, --help`	
Description: Show information about supported command line parameters on the screen and terminate module		
`-a`		`<Agent socket address>`
Description: Start **Dr.Web Daemon** in a central protection mode under control of the specified copy of **Dr.Web Agent**		
`-ini`		`<path to file>`
Description: Module must use the specified configuration file		

Short case	Extended case	Arguments
	`--foreground`	`<yes|no>`
Description: Operation mode of **Dr.Web Daemon**. If `yes` is specified, **Dr.Web Daemon** will be a foreground process. In other case (`no`) it will be a background process		
	`--check-only`	`<command line parameters for checking>`
Description: Check of a **Dr.Web Daemon** configuration correctness at start. If any command line parameters are specified, correctness of their values also will be checked		
	`--only-key`	
Description: **Dr.Web Daemon** should receive from **Dr.Web Agent** at start only the license key file		

Running Dr.Web Daemon

When **Dr.Web Daemon** is started with default settings, the following actions are performed:

- Configuration file is located and loaded. If configuration file is not found, then loading process terminates. Path to configuration file can be specified at startup by the command line parameter `-ini: {path/to/your/drweb32.ini}` or default value (`%etc_dir/drweb32.ini`) can be used. At start several configuration parameters get validated, and if parameter value is incorrect, default value is applied;
- Log file is created. User account used by **Dr.Web Daemon** must have appropriate privileges to write to the log file directory. Users do not have write permission for the default log directory (`/var/log/`). If **`User`** parameter is specified, you must also redefine **`LogFileName`** parameter and provide alternative log file location;
- Key file is loaded from the location specified in configuration file. If the key file is not found, then loading process

terminates;

- If **User** parameter is specified, **Dr.Web Daemon** will offer to create an appropriate user account (default value is `drweb`) and to use it with the permissions provided;
- **Dr.Web Engine** (`drweb32.dll`) is loaded. If **Dr.Web Engine** is damaged or not found (because of some errors in configuration file), then loading process terminates;
- Virus databases are loaded in arbitrary sequence from the location specified in configuration file. If virus databases are damaged or absent, loading process proceeds;
- **Dr.Web Daemon** enters daemon mode, so all information about loading problems can not be output to console and is written to log file;
- Socket for interaction between **Dr.Web Daemon** and other **Dr.Web for Novell Storage Services** solution modules is created. When TCP-sockets are used, there can be several connections (loading continues if at least one connection is established). When UNIX socket is used, **Dr.Web Daemon**'s user account must have appropriate privileges to read from the directory containing this socket and write to it. User accounts for modules must have execution access to the directory itself and write and read access to the socket file. Users do not have write permission for the default socket directory (`/var/run/`). If **User** parameter is specified, you must also redefine **Socket** parameter and provide alternative path to socket file. If UNIX socket was not created, then loading then loading process terminates;
- PID-file with **Dr.Web Daemon** PID information and transport addresses is created. User account used by **Dr.Web Daemon** must have appropriate privileges to write to the directory containing PID-file. Users do not have write permission for the default socket directory (`/var/run/`). If **User** parameter is specified, you must also redefine **PidFile** parameter and provide alternative path to PID-file. If PID-file was not created, then loading then loading process terminates.

Dr.Web Daemon Testing and Diagnostics

If no problems have occurred during initialization, **Dr.Web Daemon** is ready to work. To make sure that daemon have initialized correctly, issue command

```
$ netstat -a
```

and check whether necessary sockets have been created.

TCP sockets:

```
. . .
Active Internet connections (servers and established)
Proto   Recv-Q   Send-Q   Local Address   Foreign Address
State
. . .
tcp     0        0        localhost:3000  *:*
LISTEN
. . .
```

Unix socket:

```
. . .
Active UNIX domain sockets (servers and established)
Proto   RefCnt   Flags   Type    State      I-Node   Path
. . .
unix    0       [ ACC ]  STREAM  LISTENING  1127     %
var_dir/.daemon
. . .
```

If necessary sockets are missing from this list, there were problems

with **Dr.Web Daemon** initialization.

To run functional test and obtain service information use console client for **Dr.Web Daemon** (**drwebdc**).

TCP sockets:

```
$ drwebdc -nHOSTNAME -pPORTNUM -sv -sb
```

Unix socket:

```
$ drwebdc -uSOCKETFILE -sv -sb
```

It should output report to the console similar to this:

```
- Version: DrWeb Daemon 6.00
- Loaded bases:
Base /var/drweb/bases/drwtoday.vdb
contains 5 records.
Base /var/drweb/bases/drw60003.vdb
contains 409 records.
Base /var/drweb/bases/drw60002.vdb
contains 543 records.
Base /var/drweb/bases/drwebase.vdb
contains 51982 records.
Base /var/drweb/bases/drw60001.vdb
contains 364 records.
Total 53303 virus-finding records.
```

If that did not happen, run extended diagnostics.

For TCP socket:

```
$ drwebdc -nHOSTNAME -pPORTNUM -sv -sb -v
```

For UNIX socket:

```
$ drwebdc -uSOCKETFILE -sv -sb -v
```

More detailed report can identify the problem:

```
dwlib: fd: connect() failed - Connection refused
dwlib: tcp: connecting to 127.0.0.1:3300 - failed
dwlib: cannot create connection with a DrWeb daemon
ERROR: cannot retrieve daemon version
Error -12
```

You can test **Dr.Web Daemon** with special **eicar.com** program included in the installation package. Use your text editor of choice to transform `readme.eicar` into `eicar.com` (see instructions within the file).

For TCP-socket:

```
$ drwebdc -n<HOST> -p<PORT> eicar.com
```

For UNIX socket:

```
$ drwebdc -u<SOCKETFILE> eicar.com
```

It should output the following result:

```
Results: daemon return code 0x20
(known virus is found)
```

If that did not happen, check **Dr.Web Daemon** log file to see whether the file has been scanned. If the file has not been scanned, run extended diagnostic (see above).

If file was scanned successfully, **Dr.Web Daemon** is ready to operate.

When scanning very large archives, some issues with timeout expiration may occur. To fix this, increase values of `FileTimeout` and `SocketTimeout` parameters.

Please note that **Dr.Web Daemon** cannot scan files larger than **2 Gbytes**. Such files will not be sent to **Dr.Web Daemon** for scanning by clients.

Scanning Modes

Dr.Web Daemon can scan for viruses

- chunks of data received from socket (**remote scanning mode**);
- files on disk (**local scanning mode**).

In the **remote mode** client sends data to be scanned to **Dr.Web Daemon** through socket. **Dr.Web Daemon** can scan both anonymous memory and memory mapped to file system, the only difference will be logging information. This mode enables scanning files without read access but is less efficient than local scanning.

Local scanning mode is easier to use and provides better performance since client sends to **Dr.Web Daemon** only path to file to be scanned, not the whole file. Because clients can be located on different computers, the path must be specified with regard to the actual location of **Dr.Web Daemon**.

Local scan mode requires careful management of user privileges. **Dr.Web Daemon** must have read access to each file to be scanned. To perform **Cure** and **Delete** actions for files in mailboxes, you must also permit write access.

Please note, that properly configured system does not require **Dr.Web Daemon** to use root privileges.

Signal processing

Dr.Web Daemon can receive and process the following signals:

- `SIGHUP` – reload configuration file;
- `SIGTERM` – request correct **Dr.Web Daemon** termination;
- `SIGKILL` – force **Dr.Web Daemon** termination (if any problems have emerged);
- `SIGUSR1` – processes pool statistics to the log file.

Please note that `SIGUSR1` signal must be sent to parent process only, because child processes will be terminated in case receive of `SIGUSR1`.

Log Files and Statistics

Daemon Log

Since **Dr.Web Daemon** is a background program, information on its operation can only be obtained through log file. Log file contains details on processing of all scanning request sent to **Dr.Web Daemon**. You can set the log file location in parameter **`LogFileName`**.Alternatively, you can use **`syslog`** service to handle the logging (by specifying value `syslog` for parameter **`LogFileName`**).

Also, logging can be split to different files depending on **Dr.Web Daemon**'s client by setting **`ClientsLogs`** parameter. You can use this option to set up different **Dr.Web Daemon** log files for different clients that use the same **Dr.Web Daemon** to process their scanning requests.

Regardless of **`ClientsLogs`** parameter, if **Dr.Web Daemon** recognizes its client, scanning results will begin with prefix that identifies the client. Following prefixes are possible:

- `<web>` – **Dr.Web ICAPD**;
- `<smb_spider>` – **Dr.Web Samba SpIDer**;
- `<mail>` – **Dr.Web MailD**;
- `<drwebdc>` – console client for **Dr.Web Daemon**;
- `<kerio>` – **Dr.Web for Kerio Internet Gateways**;
- `<lotus>` – **Dr.Web for IBM Lotus Domino**.

On **FreeBSD** operating system information output to console by **Dr.Web Daemon** may be intercepted by **syslog** service and logged character-by-character. This is an issue of **FreeBSD** logging service that manifest itself if in `syslog.conf` configuration file logging level is set as `*.info`.

Processes pool statistics

Statistics on pool used for processing scanning request is output to the log file on receiving `SIGUSR1` signal (signal must be sent only to parent process, if a child process receives `SIGUSR1`, it will terminate) and on termination of **Dr.Web Daemon**.

A output of statistics of processes pool is controlled by **stat** value (`yes` or `no`), specified in the **ProcessesPool** parameter. Collected statistics is not aggregated. Each time the saved record contain statistics about pool state between previous and current moment of saving.

Example of pool statistics record output:

```
Fri Oct 15 19:47:51 2010 processes pool
statistics: min = 1 max = 1024 (auto)
freetime = 121 busy max = 1024 avg =
50.756950 requests for new process = 94
(0.084305 num/sec) creating fails = 0 max
processing time = 40000 ms; avg = 118646 ms
curr = 0 busy = 0
```

where:

- `min` – minimal number of processes in the pool;
- `max` – maximal number of processes in the pool;
- `(auto)` – displayed if limits on number of processes in the pool are determined automatically;
- `freetime` – maximum idle time for process in the pool;
- `busy max` – maximum number of simultaneous busy processes, `avg` - average number of simultaneous busy processes;

- `requests for new process` – number of request for new process creation (frequency of requests per second is displayed in parenthesis);
- `creating fails` – number of failed attempts of new process creation (failures are usually caused by insufficient resources);
- `max processing time` – maximum time for processing a single scanning request;
- `avg` – average time for processing a single scanning request;
- `curr` – current number of all processes in the pool;
- `busy` – current number of busy processes in the pool.

Configuration

Dr.Web Daemon can be used with default settings, but it could be convenient to configure it according to your specific requirements. **Daemon** settings are stored in `[Daemon]` section of the configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory. To use another configuration file specify full path to it with command-line option.

[Daemon]

EnginePath = {path to file}	Location of `drweb32.dll` module (anti-virus engine **Dr.Web Engine**). This parameter is also used by the **Dr.Web Updater**. Default value: **EnginePath** = %bin_dir/lib/drweb32.dll
VirusBase = {list of files (masks)}	Masks for loading virus databases. This parameter is also used by **Dr.Web Updater**. Multiple values are allowed (separated by commas). By default, virus databases files has a .

	vdb extension Default value: **VirusBase** = %var_dir/bases/*.vdb
UpdatePath = {path to directory}	This parameter is used by the **Dr.Web Updater** (update.pl) and is mandatory. Default value: **UpdatePath** = %var_dir/updates/
TempPath = {path to directory}	Directory where anti-virus engine **Dr.Web Engine** puts temporary files. It is used when system has insufficient memory or to unpack certain types of archives. Default value: **TempPath** = %var_dir/spool/
Key = {path to file}	Key file location (license or demo). Please note that **Dr.Web Daemon** and **Dr.Web Scanner** can have different license key files. In this case you must change the value of this parameter correspondingly. **Dr.Web Daemon** can use several license key files simultaneously. For each of them **Key** parameter value in [Daemon] section of drweb32.ini file must be specified. In this case **Dr.Web Daemon** tries to combine all license permissions from all available license key files. By default, key files has a .key extension Default value: **Key** = %bin_dir/drweb32.key
OutputMode = {Terminal | Quiet}	Output mode: • Terminal – console output

	• `Quiet` – no output Default value: **`OutputMode`** = `Terminal`
`RunForeground` = `{logical}`	Allows to disable or enable daemon mode for the **Dr.Web Dr.Web Daemon**. With `Yes` value specified **Dr.Web Daemon** will run as foreground process. This parameter can be used for certain monitoring utilities (for example, `daemontools`). Default value: **`RunForeground`** = `No`
`User` = `{text value}`	User which privileges will be used by the **Dr.Web Daemon**. It is strongly recommended to create separate **`drweb`** user account, which will be used by the **Dr.Web Daemon** and filters. It is not recommended to run **Dr.Web Daemon** with `root` privileges, even though it may take less time to set up. This parameter cannot be changed when reloading configuration using `SIGHUP`. Default value: **`User`** = `drweb`
`PidFile` = `{path to file}`	File to store **Dr.Web Daemon**'s PID and UNIX socket (if it is enabled by **`Socket`** parameter) or port number (if TCP socket is enabled by **`Socket`** parameter). If more than one **`Socket`** parameter is specified, this file will contain information on all the sockets (one per line). This file is created every time **Dr.Web Daemon** starts.

	Default value: **PidFile** = %var_dir/run/drwebd.pid
BusyFile = {path to file}	File where **Dr.Web Daemon**'s busy flag is stored. This file is created by a **Dr.Web Daemon** child process upon a receipt of the scan command and is removed after successful execution of the command. Filenames created by each **Dr.Web Daemon** child process are appended by a dot and ASCIIZ representation of PID (for example, /var/run/drwebd.bsy.123456). Default value: **BusyFile** = %var_dir/run/drwebd.bsy
ProcessesPool = {process pool settings}	Dynamic process pool settings. First number of processes in the pool must be specified: • auto – number of processes will be set automatically depending on system load; • N – unsigned integer number. Pool will have at least N active processes, additional processes will be created if necessary; • N-M – integer unsigned numbers, M>=N. Pool will have at least N active processes, additional processes will be created if necessary, but maximum total number of processes cannot exceed M. Then, optional secondary parameters may be specified: • **timeout** = {time in seconds} — timeout for closing an inactive

	process. This parameter does not affect first `N` processes which await requests continually. • **`stat`** = `{yes|no}` — statistics for processes in a pool. If `yes`, it is saved to the log file each time `SIGUSR1` system signal is received. • **`stop_timeout`** = `{time in seconds}` — maximum waiting period for stopping a working process. Default value: **`ProcessesPool`** = `auto`, **`timeout`** = `120`, **`stat`** = `no`, **`stop_timeout`** = `1`
`OnlyKey` = `{logical}`	Only license key file will be received from the **Dr.Web Agent**. Local configuration file will be used for all the settings. If the value of this parameter is `No`, and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Daemon** will sends work statistics to **Dr.Web Agent** (sending of information will be carried out after scanning of each file). Default value: **`OnlyKey`** = `No`
`ControlAgent` = `{address}`	**Dr.Web Agent**'s socket address. **Example**: **`ControlAgent`** = `inet:4040@127.0.0.1,local:%var_dir/ipc/.agent` **Dr.Web Daemon** receives from the **Dr.Web Agent** license key file (and configuration information in case **`OnlyKey`** = `No`. Also in this case socket is used for sending **Dr.Web Daemon**'s work statistics

	to **Dr.Web Agent**). Default value: **ControlAgent** = local:% var_dir/ipc/.agent
MailCommand = {string}	Shell command used by the **Dr.Web Daemon** and the **Dr.Web Updater** for sending notifications and information bulletins on new updates to the user (administrator) via e-mail. If the period before the key file (or one of the key files) expiration is less, than is specified in **NotifyPeriod** parameter, **Dr.Web Daemon** starts sending out notifications every time system starts, restarts or reboots. Default value: **MailCommand** = "/usr/sbin/ sendmail -i -bm -f drweb -- root"
NotifyPeriod = {numerical value}	This parameter value specifies how many days should be left before license expiration for the **Dr.Web Daemon** to start sending notifications of license renewal. If parameter value is set to 0, **Dr.Web Daemon** starts sending out notifications immediately after key file expires. Default value: **NotifyPeriod** = 14
NotifyFile = {path to file}	File with a timestamp of last notification of license renewal. Default value: **NotifyFile** = %var_dir/.notify

NotifyType = {Ever | Everyday | Once}	Frequency of license expiration notifications. • Once – notification is sent only once. • Everyday – notification is sent daily. • Ever – notification is sent with every **Dr.Web Daemon** restart and every database update. Default value: **NotifyType** = Ever
FileTimeout = {numerical value}	Maximum time (in seconds) allowed for the **Dr.Web Daemon** to perform a scan of one file. If parameter value is set to 0, time to scan of one file is unlimited. Default value: **FileTimeout** = 30
StopOnFirstInfected = {logical}	Enables or disables stopping file scan upon detection of the first virus. Default value: **StopOnFirstInfected** = No
ScanPriority = {signed numerical value}	Priority of **Dr.Web Daemon** process. Value must be in the following range: -20 (highest priority) to 19 (lowest priority for **Linux**) or 20 (lowest priority for **FreeBSD** and **Solaris**). Default value: **ScanPriority** = 0
FilesTypes = {list of file extensions}	File types to be checked "by type", i.e. when **ScanFiles** parameter (explained below) has ByType value. "*" and "?" wildcard characters are allowed.

	Default value: **FilesTypes** = EXE, COM, SYS, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, VXD, 386, DLL, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, AR?, ZIP, R??, PP?, OBJ, LIB, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SHS, SHB, PIF, SO, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, BMP, MPP, OCX, DVB, CPY, MSG, EML
FilesTypesWarnings = {logical}	Notify about files of unknown types Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Scan only files with extensions specified in **FileTypes** parameter (value ByType) or all files (value All). Value ByType of this parameter can be used only in **local scan** mode (in other modes always used only All value). In mailboxes are always scanned all files (independent of **ScanFiles** parameter value). Default value: **ScanFiles** = All
CheckArchives = {logical}	Enables or disables checking of files in archives (RAR, ARJ, TAR, GZIP, CAB and others). Default value: **CheckArchives** = Yes
CheckEMailFiles =	Enables or disables checking mailbox files.

	Default value: **CheckEMailFiles** = Yes
ExcludePaths = {list of path | file masks}	Masks for files to be skipped during scanning. Multiple values are allowed (separated by commas). Default value: **ExcludePaths** = /proc,/sys,/dev
FollowLinks = {logical}	Determine if **Dr.Web Daemon** will follow symbolic links during scanning. Default value: **FollowLinks** = No
RenameFilesTo = {mask}	Mask for renaming infected or suspicious files if action Rename is specified. Default value: **RenameFilesTo** = #??
MoveFilesTo = {path to directory}	Path to the **Quarantine** directory. Default value: **MoveFilesTo** = %var_dir/infected/
BackupFilesTo = {path to directory}	Directory for backup copies of infected files made if requested action was Cure. Default value: **BackupFilesTo** = %var_dir/infected/
LogFileName = {syslog | file name}	Log file name. You can specify syslog as log file name and logging will be carried out by syslogd system service.

	In this case you must also specify **SyslogFacility** and **SyslogPriority** parameters. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Logging priority (log verbosity level) when syslogd system service is used. Following levels are allowed: • Error • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
LimitLog = {logical}	Enables or disables limit for log file size (if **LogFileName** value is not specified to syslog). If limit is enabled, **Dr.Web Daemon** will check the size of log file on startup or receiving the SIGHUP signal. If log file size is greater then MaxLogSize value, log file will overwritten with empty file and logging will begin from scratch. Default value: **LimitLog** = No
MaxLogSize = {numerical value}	Maximum log file size in Kbytes. Used only with **LimitLog** = Yes.

	Set this parameter value to `0` if you do not want log file to be unexpectedly modified at start up. Default value: **MaxLogSize** = 512
LogScanned = {logical}	Enables or disables logging of information about all scanned objects regardless whether they are infected or not. Default value: **LogScanned** = Yes
LogPacked = {logical}	Enables or disables logging of additional information about files packed with DIET, PKLITE and other utilities. Default value: **LogPacked** = Yes
LogArchived = {logical}	Enables or disables logging of additional information about files archived with various archiving utilities. Default value: **LogArchived** = Yes
LogTime = {logical}	Enables or disables logging of time for each record. Parameter is not used if **LogFileName** = syslog. Default value: **LogTime** = Yes
LogProcessInfo = {logical}	Enables or disables logging of every scanning process PID and filter address (host name or IP address) from which scanning has been activated. This data is put before each record. Default value: **LogProcessInfo** = Yes

RecodeNonprintable = {logical}	Non-printable characters output mode for a given terminal. Default value: **RecodeNonprintable** = Yes
RecodeMode = {Replace | QuotedPrintable}	Decoding mode for non printable characters if **RecodeNonprintable** = Yes. When **RecodeMode** = Replace all non-printable characters are substituted with **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable all non printable characters are converted to quoted printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets character for replacing non-printable characters if **RecodeMode** = Replace. Default value: **RecodeChar** = "?"
Socket = {address list}	List of sockets to be used for communication with **Dr.Web Daemon** (separated by a commas). **Example**: Socket = inet:3000@127.0.0.1, local:%var_dir/.daemon Also you can specify socket address in PORT [interfaces] | FILE [access] format. For a TCP socket, specify decimal port number (PORT) and the list of interface names or IP addresses for incoming requests (interfaces).

	Example:
	`Socket = 3000 127.0.0.1, 192.168.0.100`
	For UNIX sockets, specify socket name (FILE) and access permissions in octal form (access).
	Example:
	`Socket = %var_dir/.daemon`
	Number of values of **Socket** parameter is not limited. **Dr.Web Daemon** will work with all correctly described sockets.
	To enable connections on all available interfaces set 3000 0.0.0.0 as a value for this parameter.
	Default value:
	`Socket = %var_dir/run/.daemon`

SocketTimeout = {numerical value}	Maximum time (in seconds) allowed for transferring data through socket (file scanning time is not included). If parameter value is set to `0`, this time is unlimited. Default value: `SocketTimeout = 10`
ClientsLogs = {string list}	Enables splitting the log files. If during communication with **Dr.Web Daemon** client uses the option to transfer its ID, log file will be substituted with the file specified in this parameter. Descriptions of log files are delimited by commas or whitespaces. If more than six values are set, configuration file is considered invalid. The log files are defined in the following way:

	`<client name1>:<path to file>, <client name2>:<path to file>`
	Client name may be one of the following:
	• `web` — **Dr.Web ICAPD**;
	• `smb_spider` — **Dr.Web Samba SpIDer**;
	• `mail` — **Dr.Web MailD**;
	• `drwebdc` — console client for **Dr.Web Daemon**;
	• `kerio` — **Dr.Web for Kerio Internet Gateways**;
	• `lotus` — **Dr.Web for IBM Lotus Domino**.
	Example:
	`drwebdc:/var/drweb/log/drwebdc.log,`
	`smb:syslog,`
	`mail:/var/drweb/log/drwebmail.log`

	Default value:
MaxBasesObsolescencePeriod = {numerical value}	Period in hours after last update, during which virus databases are considered up-to-date. When this period is over, notification that databases are obsolete will be output to console. If value is set to 0, database obsolescence will not be checked.
	Default value: **MaxBasesObsolescencePeriod** = 24

The following parameters can be used to reduce scanning time in archive files (some objects in archives will not be checked). Actions applied to skipped archives are determined in `ArchiveRestriction` parameter of the corresponding modules.

`MaxCompressionRatio` `= {numerical value}`	Maximum compression ratio, i.e. ratio of unpacked file size to packed file size. Parameter can take only natural values. If the ratio exceeds specified value, file will not be extracted and therefore will not be checked. E-mail message with such file is considered "mail bomb". If value 0 is specified, compression ratio will not be checked. Default value: **`MaxCompressionRatio`** `= 500`
`CompressionCheckThreshold` `= {numerical value}`	Minimum size of the file inside an archive in Kbytes, for which compression ratio check will be performed (if it is specified by the **`MaxCompressionRatio`** parameter). If value 0 is specified, check will not be performed. Default value: **`CompressionCheckThreshold`** `= 1024`
`MaxFileSizeToExtract` `= {numerical value}`	Maximum unpacked size for the file in an archive in Kbytes. If unpacked size exceeds specified value the archive will not be scanned. E-mail letter with such file is considered "mail bomb". If value 0 is specified, maximum unpacked size is unlimited. Default value: **`MaxFileSizeToExtract`** `= 40960`

MaxArchiveLevel = {numerical value}	Maximum allowed archive nesting level. If archive nesting level exceeds specified value, it will not be scanned. E-mail message with such file is considered "mail bomb". If value 0 is specified, depth of nesting level will not be checked Default value: **MaxArchiveLevel** = 8
MessagePatternFileName = {path to file}	Path to template for message about license expiration. You can define expiration message according to your requirements. You can use variables that will be substituted for the following values: • $EXPIRATIONDAYS — number of day left until the license would expire; • $KEYFILENAME — path to license key file; • $KEYNUMBER — license number; • $KEYACTIVATES — license activation date; • $KEYEXPIRES — license expiration date. If there is no user-defined template, standard message in English will be used. Default value: **MessagePatternFileName =** %etc_dir/templates/drwebd/msg.tmpl
MailTo = {email address}	Administrator email address to send messages about license expiration, virus databases obsolescence, etc.

	Default value: `MailTo =`

Command Line Scanner Dr.Web

Command line scanner **Dr.Web Scanner** serves for detection and neutralization of malware on the local machine. Component is presented by a module **drweb**.

Dr.Web Scanner at start checks the specified files and boot records of the specified disks. For anti-virus checking and curing **Dr.Web Scanner** uses the **Dr.Web Engine** and virus bases, but doesn't use the resident module **Dr.Web Daemon** (work is made independent of it).

Command Line Parameters

You can run **Dr.Web Scanner** with the following command:

```
$ %bin_dir/drweb <path> [parameters]
```

where `<path>` – is the path or paths to scanned directories or the mask for checked files. If in startup path is specified with following prefix: `disk://<path to device file>` then boot sector of appropriate device will be checked and cured, if necessary. When **Dr.Web Scanner** is started only with `<path>` argument without any parameters specified, it scans the specified directory using the default set of parameters. Details about `<path>` parameter specification you can see below.

In the following example user home directory is being checked:

```
$ %bin_dir/drweb ~
```

When scanning is finished **Dr.Web Scanner** displays all found infected and suspicious files in the following manner:

```
/path/file infected [virus] VIRUS_NAME
```

After presenting information about infected or suspicious files, **Dr. Web Scanner** outputs summary report in the following manner:

```
Report for "/opt/drweb/tmp":
Scanned         : 34/32     Cured       : 0
Infected        : 5/5       Removed     : 0
Modifications   : 0/0       Renamed     : 0
Suspicious      : 0/0       Moved       : 0
Scan time       : 00:00:02 Scan speed : 5233
KB/s
```

Numbers divided by slash "/" mean: the first one – total number of files, the second one – number of files in archives.

You can use `readme.eicar` file included in the distribution package to test **Dr.Web Scanner**. Open this file in your text editor of choice and follow the instructions contained in the file to transform it into `eicar.com` program.

When you check it with **Dr.Web Scanner**, it should output the following message:

```
%bin_dir/doc/eicar.com infected by Eicar Test File
(Not a Virus!)
```

This program is not a virus and is used only for testing of anti-virus programs.

Dr.Web Scanner has many command-line parameters. In accordance to UNIX conventions they are separated from path by whitespace character and start with a hyphen("-"). To get complete list of parameters run **Dr.Web Scanner** with either `-?`, `-h`, or `-help` parameters.

The **Console Scanner** parameters can be divided into the following groups:

- Scan area parameters
- Diagnostics parameters
- Action parameters
- Interface parameters

Scan Area Parameters

These parameters determine where to perform a virus scan:

Parameter	Description
`-path [=] <path>`	Sets the scan path. Symbol '=' can be skipped, in this case path for scan is delimited from `-path` parameter by a white space. You can specify several paths in one `-path` parameter (paths will aggregate to one list). Also you can specify paths without `-path` parameter. If `<path>` is specified with following prefix in startup options: `disk://<path to device file>`, then boot sector (MBR) of appropriate device will be checked and cured, if necessary. Device file is a special file, placed in directory `/dev` and having name like as **`sdX`** or **`hdX`**, where **`X`** – letter of latin alphabet (`a`, `b`, `c`, ...). For example: **`hda`**, **`sda`**. So, if you want to check MBR of disk **`sda`**, specify: `disk:///dev/sda`
`-@[+]<file>`	Instructs to scan objects listed in the specified file. Add a plus '+' if you do not want the list-file to be deleted when scanning completes. List file may contain paths to directories that must be scanned regularly, or list of files to be checked only once.
`--`	Instructs to read list of objects to scan from the standard input (STDIN).
`-sd`	Sets recursive search for files to scan in subfolders.
`-fl`	Instructs to follow symbolic links to both files and folders. Links causing loops are ignored.
`-mask`	Instructs to ignore masks for filenames.

Diagnostics Parameters

These parameters determine which types of objects to scan for viruses:

Parameter	Description
-al	Instructs to scan all objects defined by scan paths regardless of their file extension and structure. This parameter is opposite in effect to the **-ex** parameter.
-ex	Instructs to search scan paths for threats presented by files of certain types and ignore objects of other types. The list of file types should be specified in the **FileTypes** variable of the configuration file. The configuration file is defined by the **-ini** parameter. By default, objects with the following file extensions are scanned: EXE, COM, DLL, SYS, VXD, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, 386, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, PP?, OBJ, LIB, PIF, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SH, SHB, SHS, SHT*, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, MPP, OCX, VS*, DVB, CPY, BMP, RPM, ISO, DEB, AR?, ZIP, R??, GZ, Z, TGZ, TAR, TAZ, CAB, LHA, LZH, BZ2, MSG, EML, 7Z, CPIO. This parameter is opposite in effect to the **-al** parameter.
-ar[d|m|r][n]	Instructs to scan contents of archives (ARJ, CAB, GZIP, RAR, TAR, ZIP, etc.), both simple (*.tar) and compressed (*.tar.bz2, *.tbz). If you do not supplement the parameter with an additional **d**, **m** or **r** modifier, **Dr.Web Scanner** only informs you about detected malicious or suspicious files in archives. Otherwise, it applies appropriate actions to avert detected threats.
-cn[d|m|r][n]	Instructs to scan contents of files containers (HTML, RTF, PowerPoint). If you do not supplement the parameter with an additional **d**, **m** or **r** modifier, **Dr.Web Scanner** only informs you about detected malicious or suspicious

Parameter	Description
	files in containers. Otherwise, it applies appropriate actions to avert detected threats.
-ml[d|m|r][n]	Instructs to scan contents of mail files. If you do not supplement the parameter with an additional **d**, **m** or **r** modifier, **Dr.Web Scanner** only informs you about detected malicious or suspicious elements of mail files. Otherwise, it applies appropriate actions to avert detected threats.
-upn	Scan executable files packed with LZEXE, DIET, PKLITE, EXEPACK with compression type output disabled
-ha	Enables heuristic analyser that help detect possible unknown threats.

For some parameters, you can use the following additional modifiers:

- Add **d** to delete objects to avert the threat
- Add **m** to move objects to **Quarantine** to avert the threat
- Add **r** to rename objects to avert the threat (that is, replace the first character of the file's extension with '#')
- Add **n** to disable output of the archive, container, mail file or packer type

If malicious objects are detected within complex objects such as archives, containers, packed or mail files, then the reaction is applied to the complex object as a whole, and not to the included malicious object only.

Action Parameters

These parameters determine which actions to apply to infected (or suspicious) objects:

Parameter	Description
-cu[d|m|r]	Defines an action to apply to infected files and boot sectors. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** cures infected objects and deletes incurable files (if another action

Parameter	Description
	is not specified in the **-ic** parameter). Otherwise, it applies appropriate action to infected curable object, and processes incurable files as specified in the **-ic** parameter.
-ic[d|m|r]	Defines an action to apply to incurable files. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-sp[d|m|r]	Defines an action to apply to suspicious files. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-adw[d|m|r|i]	Defines an action to apply to adware. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-dls[d|m|r|i]	Defines an action to apply to dialers. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-jok[d|m|r|i]	Defines an action to apply to joke programs. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-rsk[d|m|r|i]	Defines an action to apply to potentially dangerous programs. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.
-hck[d|m|r|i]	Defines an action to apply to hacktools. If you do not supplement the parameter with an additional modifier, **Dr.Web Scanner** only informs you about the threat.

Additional modifiers indicate actions that should be applied for averting threats:

- Add **d** to delete objects.
- Add **m** to move objects to **Quarantine**.
- Add **r** to rename objects, that is, replace the first character of extension with '#'.
- Add **i** to ignore threats (available for minor threats only such as adware etc), that is, apply no action and do not list such threats in the report.

If malicious objects are detected within complex objects such as archives, containers, packed or mail files, then the reaction is applied to the complex object as a whole, and not to the included malicious object only.

Interface Parameters

These parameters configure **Dr.Web Scanner** output:

Parameter	Description
`-v, -version, --version`	Instructs to output information about the product and scan engine versions and exit **Dr.Web Scanner**.
`-ki`	Instructs to output information about the license and its owner (in UTF8 encoding only).
`-go`	Instructs to run **Dr.Web Scanner** in batch mode when all questions implying answers from a user are skipped and all decisions implying a choice are taken automatically. This mode is useful for automatic scanning of files, for example, during a daily (or weekly) check of the hard drive.
`-ot`	Instructs to use the standard output (STDOUT).
`-oq`	Disables information output.
`-ok`	Instructs to list all scanned objects in the report and mark "clean" object with **Ok**.

Parameter	Description
-log=[+]<path to file>	Instructs to log **Dr.Web Scanner** operations in the specified file. The file name is mandatory to turn on logging. Add a plus '+' if you want to append the log file instead of overwriting it.
-ini=<path to file>	Instructs to use the specified configuration file. No configuration file is supplied with **Dr.Web Scanner** by default.
-lng=<path to file>	Instructs to use the specified language file. The default language is English.
-a = <Control Agent address>	Run **Dr.Web Scanner** in central protection mode.
-ni	Disables the use of the configuration file for setting up scanning options. **Dr.Web Scanner** is configured with parameters from the command line only.
-ns	Disables interruption of scanning process including the use of interruption signals (SIGINT).
--only-key	Nothing but key file is received from the Control Agent at start.

You can use hyphen «-» postfix to disable the following parameters:

-ar -cu -ha -ic -fl -ml -ok -sd -sp

For example, if you start **Dr.Web Scanner** with the following command:

```
$ drweb <path> -ha-
```

heuristic analysis (enabled by default) will be disabled.

For the `-cu`, `-ic` and `-sp` parameters, the negative form disables any action specified with additional modifiers, that is, negative form of these parameters instruct to report on detection of infected or suspicious objects, but take no actions to avert threats.

The `-al` and `-ex` parameters have no negative for, but cancel one another.

By default (if **Dr.Web Scanner** configuration was not customized and no parameters were specified) **Dr.Web Scanner** starts with the following parameters:

```
-ar -ha -fl- -ml -sd -al -ok
```

Default **Dr.Web Scanner** parameters (including scan of archives, packed files and mailboxes, recursive search, heuristic analysis, etc.) is sufficient for everyday diagnostics and can be used in typical cases. You can also use hyphen «-» postfix to disable some parameters.

Disabling scan of archives and packed files will significantly decrease antivirus protection level, because in archives (especially, self-extracting) enclosed in e-mail attachments viruses are distributed. Office documents potentially susceptible to infection with macro viruses (Word, Excel) are also dispatched via e-mail in archives and containers.

When you run **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are taken. For these actions to be performed, you must specify corresponding command line parameters explicitly.

Configuration File

Dr.Web Scanner can be used with default settings, but it could be convenient to configure it according to your specific requirements. **Dr.Web Scanner** settings are stored in configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory.

To use another configuration file, specify full path to it with command line parameter, for example:

```
$ %bin_dir/drweb -ini=%bin_dir/etc/drweb.ini
```

General principles of the **Dr.Web for Novell Storage Services** configuration files organization see in chapter Configuration files.

[Scanner]

EnginePath = {path to file}	Location of drweb32.dll module (anti-virus engine **Dr.Web Engine**). This parameter is also used by **Dr.Web Updater**. Default value: **EnginePath** = %bin_dir/lib/drweb32.dll
VirusBase = {list of file masks}	Masks for loading virus databases. This parameter is also used by **Dr.Web Updater**. Multiple values are allowed (separated by commas). By default, virus databases files has a .vdb extension Default value: **VirusBase** = %var_dir/bases/*.vdb
UpdatePath = {path to directory}	This parameter is used by **Dr.Web Updater** (update.pl) and is mandatory. Default value: **UpdatePath** = %var_dir/updates/
TempPath = {path to directory}	Directory where anti-virus engine **Dr.Web Engine** puts temporary files. It is used when system has insufficient

	memory or to unpack certain types of archives. Default value: **TempPath** = /tmp/
LngFileName = {path to file}	Language file location. By default, language files has a .dwl extension Default value: **LngFileName** = %bin_dir/lib/ru_scanner.dwl
Key = {path to file}	Key file location (license or demo). By default, key files has a .key extension Default value: **Key** = %bin_dir/drweb32.key
OutputMode = {Terminal | Quiet}	Output mode: • Terminal – console output • Quiet – no output Default value: **OutputMode** = Terminal
HeuristicAnalysis = {logical}	Enable or disable heuristic detection of unknown viruses. Heuristic analysis can detect previously unknown viruses which are not included in the virus database. It relies on advanced algorithms to determine if scanned file structure is similar to the virus architecture. Because of that heuristic analysis can produce false positives: all objects detected by this method are considered suspicious. Please send all suspicious files to **Dr.Web** through http://vms.drweb.com/sendvirus/ for checking. To send suspicious file, put it in password protected archive, include

	password in message body and attach **Dr.Web Scanner** report. Default value: **HeuristicAnalysis** = Yes
ScanPriority = {signed numerical value}	**Dr.Web Scanner** process priority. Value must be between -20 (highest priority) and 19 (**Linux**) or 20 (other UNIX-like operating systems) range. Default value: **ScanPriority** = 0
FilesTypes = {list of file extensions}	File types to be checked "by type", i.e. when **ScanFiles** parameter (explained below) has ByType value. "*" and "?" wildcard characters are allowed. Default value: FilesTypes = EXE, COM, SYS, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, VXD, 386, DLL, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, AR?, ZIP, R??, PP?, OBJ, LIB, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SHS, SHB, PIF, SO, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, BMP, MPP, OCX, DVB, CPY, MSG, EML
FilesTypesWarnings = {logical}	Notify about files of unknown types. Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Scan all files (All) or only files with extensions specified in **FileType** parameter (ByType). Value ByType of this parameter can be

	used only in **local scan** mode (in other modes always used only `All` value). In mailboxes are always scanned all files (independent of **`ScanFiles`** parameter value). Default value: **`ScanFiles`** = `All`
`ScanSubDirectories` = `{logical}`	Enable/disable scanning in subdirectories. Default value: **`ScanSubDirectories`** = `Yes`
`CheckArchives` = `{logical}`	Enables or disables checking of files in archives (RAR, ARJ, TAR, GZIP, CAB and others). Default value: **`CheckArchives`** = `Yes`
`CheckEMailFiles` = `{logical}`	Enables or disables checking mailbox files. Default value: **`CheckEMailFiles`** = `Yes`
`ExcludePaths` = `{list of path | file masks}`	Masks for files to be skipped during scanning. Multiple values are allowed (separated by commas). Default value: **`ExcludePaths`** = `/proc,/sys,/dev`
`FollowLinks` = `{logical}`	Determine if **Dr.Web Scanner** will follow symbolic links during scanning. Default value: **`FollowLinks`** = `No`

RenameFilesTo = {mask}	Mask for renaming infected or suspicious files if action Rename is specified. Default value: **RenameFilesTo** = #??
MoveFilesTo = {path to directory}	Path to the **Quarantine** directory. Default value: **MoveFilesTo** = %var_dir/infected/
EnableDeleteArchiveAction ={logical}	Enables or disables action Delete for multipart objects (archives, mailboxes, HTML pages) if they contain infected files. Please note: with this parameter enabled whole multipart object will be deleted (archive, mailbox, etc.), not just infected file or message. Use this option carefully! Default value: **EnableDeleteArchiveAction** = No
InfectedFiles = {action}	Sets action when infected file is found: Report, Cure, Delete, Move, Rename, Ignore. Delete, Move and Rename actions, specified for archives, containers and mailboxes containing infected files, are applied to the whole archive, container or mailbox! Default value: **InfectedFiles** = Report
SuspiciousFiles = {action}	Sets action when suspicious file is found: Report, Delete, Move, Rename, Ignore.

	Default value: **SuspiciousFiles** = Report
IncurableFiles = {action}	Sets action when infected file cannot be cured (should be used only if **InfectedFiles** = Cure): Report, Delete, Move, Rename, Ignore. Default value: **IncurableFiles** = Report
ActionAdware = {action}	Sets action when adware is found: Report, Delete, Move, Rename, Ignore. Default value: **ActionAdware** = Report
ActionDialers = {action}	Sets action when dialer is found: Report, Delete, Move, Rename, Ignore. Default value: **ActionDialers** = Report
ActionJokes = {action}	Sets action when joke program is found: Report, Delete, Move, Rename, Ignore. Default value: **ActionJokes** = Report
ActionRiskware = {action}	Sets action when potentially dangerous program is found: Report, Delete, Move, Rename, Ignore. Default value: **ActionRiskware** = Report

ActionHacktools = {action}	Sets action when hacking program is found: Report, Delete, Move, Rename, Ignore. Default value: **ActionHacktools** = Report
ActionInfectedMail = {action}	Sets action when infected file is found in mailbox: Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedMail** = Report
ActionInfectedArchive = {action}	Sets action when infected file is found in archive: Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedArchive** = Report
ActionInfectedContainer = {action}	Sets action when infected file is found in container (OLE, HTML, PowerPoint and etc.): Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedContainer** = Report

Logging parameters:

LogFileName = {syslog | file name}	Log file name. You can specify syslog as log file name and logging will be carried out by syslogd system service. In this case you must also specify

	SyslogFacility and **SyslogPriority** parameters. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Logging priority (log verbosity level) when syslogd system service is used. Following levels are allowed: • Error • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
LimitLog = {logical}	Enables or disables limit for log file size (if **LogFileName** value is not specified to syslog). With this parameter enabled, **Dr.Web Scanner** will be checking log file size at startup. If log file size exceeds **MaxLogSize** parameter value, log file content will be erased and logging will start from scratch. Default value: **LimitLog** = No
MaxLogSize = {numerical value}	Maximum log file size in Kbytes. Used only with **LimitLog** = Yes.

	Set this parameter value to `0` if you do not want log file to be unexpectedly modified at start up. Default value: **`MaxLogSize`** = `512`
`LogScanned` = `{logical}`	Enables or disables logging of information about all scanned objects regardless whether they are infected or not. Default value: **`LogScanned`** = `Yes`
`LogPacked` = `{logical}`	Enable/disable logging of additional information about files packed with DIET, PKLITE and other utilities. Default value: **`LogPacked`** = `Yes`
`LogArchived` = `{logical}`	Enables or disables logging of additional information about files archived with various archiving utilities. Default value: **`LogArchived`** = `Yes`
`LogTime` = `{logical}`	Enables or disables logging of time for each record. Parameter is not used if **`LogFileName`** = `syslog`. Default value: **`LogTime`** = `Yes`
`LogStatistics` = `{logical}`	Enable/disable logging of scan statistics. Default value: **`LogStatistics`** = `Yes`
`RecodeNonprintable` = `{logical}`	Non-printable characters output mode for a given terminal.

	Default value: **RecodeNonprintable** = Yes
RecodeMode = {Replace | QuotedPrintable}	Decoding mode for non printable characters if **RecodeNonprintable** = Yes. When **RecodeMode** = Replace all non-printable characters are substituted with **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable all non printable characters are converted to quoted printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets character for replacing non-printable characters if **RecodeMode** = Replace. Default value: **RecodeChar** = "?"

Following parameters can be used to reduce scanning time in archive files (some objects in archives will not be checked).

MaxCompressionRatio = {numerical value}	Maximum compression ratio, i.e. ratio of unpacked file size to packed file size. If the ratio exceeds specified value, file will not be extracted and therefore will not be checked. Parameter can take only natural values. E-mail message with such file is considered "mail bomb". If value 0 is specified, compression ratio will not be checked Default value: **MaxCompressionRatio** = 5000

CompressionCheckThreshold = {numerical value}	Minimum size of file inside archive in Kbytes, for which compression ratio check will be performed (if it is specified by **MaxCompressionRatio** parameter). Default value: **CompressionCheckThreshold** = 1024
MaxFileSizeToExtract = {numerical value}	Maximum unpacked size for file in archive in Kbytes. If unpacked size exceed specified value it will not be scanned. E-mail letter with such file is considered "mail bomb". Default value: **MaxFileSizeToExtract** = 500000
MaxArchiveLevel = {numerical value}	Maximum allowed archive nesting level. If archive nesting level exceeds specified value, it will not be scanned. E-mail message with such file is considered "mail bomb". If value 0 is specified, depth of nesting level will not be checked Default value: **MaxArchiveLevel** = 8
MaximumMemoryAllocationSize = {numerical value}	Size of maximum amount of memory consumption allowed for scanning one file (in Mbytes). If value is set to 0, memory allocation will not be limited. Default value: **MaximumMemoryAllocationSize** = 0

ScannerScanTimeout = {numerical value}	Maximum time period allowed for scanning one file (in seconds). If value is set to 0, scanning time will not be limited. Default value: **ScannerScanTimeout** = 0
MaxBasesObsolescencePeriod = {numerical value}	Maximal period after last update during which virus databases are considered up-to-date (in hours). When this period is over, notification that databases are obsolete will be output to console. If value is set to 0, database obsolescence will not be checked. Default value: **MaxBasesObsolescencePeriod** = 24
ControlAgent = {address}	**Dr.Web Agent** socket address. **Example**: **ControlAgent** = inet:4040@127.0.0.1,local:%var_dir/ipc/.agent **Dr.Web Scanner** receives from the **Dr.Web Agent** license key file (and configuration information in case **OnlyKey** = No. Also in this case socket is used for sending **Dr.Web Scanner**'s work statistics to **Dr.Web Agent**). Default value: **ControlAgent** = local:%var_dir/ipc/.agent
OnlyKey = {logical}	Enable receiving only license key file from **Dr.Web Agent** without configuration information. **Dr.Web Scanner** will use local

	configuration file. If this parameter value is set to `No`, and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Scanner** will also be sending to **Dr.Web Agent** statistics on scanned files (sending of information will be carried out after scanning of each file). Default value: **`OnlyKey`** = `No`

Running Dr.Web Scanner

You can run **Dr.Web Scanner** with command

```
$ %bin_dir/drweb
```

If `%bin_dir` directory is added to `PATH` environment variable, you can run **Dr.Web Scanner** from any directory only by typing "`drweb`". However, doing so (as well as making a symbolic link to **Dr.Web Scanner** executable file in directories like `/bin/`, `/usr/bin/`, etc.) is not recommended for security reasons.

Dr.Web Scanner can be run with either root or user privileges. In the last case virus scanning can be only performed in directories, where user has read access, and infected files will be cured only in directories, where user has write access (usually it is user home directory, `$HOME`). Also, there are other restrictions when **Dr.Web Scanner** is started with user privileges, for example, on moving and renaming infected files.

When **Dr.Web Scanner** is started, it displays program name, platform name, program version number, release date and contact information. It also shows user registration information and statistics, list of virus databases and installed updates:

```
Dr.Web (R) Scanner for Linux, v6.0.1 (February 19,
2010)
Copyright (c) Igor Daniloff, 1992-2010
Support service: http://support.drweb.com/
To purchase: http://buy.drweb.com/
Program version: 6.0.0.10060 <API:2.2>
Engine version: 6.0.0.9170 <API:2.2>
Loading /var/drweb/bases/drwtoday.vdb - Ok, virus
records: 1533
Loading /var/drweb/bases/drw60012.vdb - Ok, virus
records: 3511
-------------------------------------------
Loading /var/drweb/bases/drw60000.vdb - Ok, virus
records: 1194
Loading /var/drweb/bases/dwn60001.vdb - Ok, virus
records: 840
Loading /var/drweb/bases/drwebase.vdb - Ok, virus
records: 78674
Loading /var/drweb/bases/drwrisky.vdb - Ok, virus
records: 1271
Loading /var/drweb/bases/drwnasty.vdb - Ok, virus
records: 4867
Total virus records: 538681
Key file: /opt/drweb/drweb32.key
Key file number: XXXXXXXXXX
Key file activation date: XXXX-XX-XX
Key file expiration date: XXXX-XX-XX
```

After displaying this report **Dr.Web Scanner** terminates. In order to scan for viruses or neutralize detected threats you must specify additional command-line parameters.

By default **Dr.Web Scanner** starts with the following parameters:

```
-ar -ha -fl- -ml -sd -al -ok
```

These parameters are optimal for thorough anti-virus protection and can be used in most typical cases. If any parameters is not required, you can disable it with "–" postfix as described above.

Disabling scan of archives and packed files will significantly decrease anti-virus protection level, because viruses are often distributed in archives (especially, self-extracting), enclosed in e-mail attachments. Office documents potentially susceptible to infection with macro viruses (Word, Excel) are also dispatched via e-mail in archives and containers.

When you run **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are taken. For these actions to be performed, you must specify corresponding command line parameters explicitly.

Following actions are recommended:

- `cu` – cure infected files and system areas without deleting, moving or renaming infected files;
- `icd` – delete incurable files;
- `spm` – move suspicious files;
- `spr` – rename suspicious files.

When **Dr.Web Scanner** is started with `cu` action specified, it will try to restore the original state of infected object. It is possible only if detected virus is a known virus, and cure instructions for it are available in virus database, though even in this case cure attempt may fail if infected file is seriously damaged by a virus.

If infected files are found inside archives they will not be cured, deleted, moved or renamed. To cure such files you must manually unpack archives to the separate directory and instruct **Dr.Web Scanner** to check it.

When **Dr.Web Scanner** is started with action `icd` specified, it will remove all infected files from disk. This option is suitable for incurable (irreversibly damaged by virus) files.

`spr` action makes **Dr.Web Scanner** replace file extension with

another extension (`*.#??` by default, i.e. first extension character is replaced with "#" character). Enable this parameter for files for other operating systems detected heuristically as suspicious. Renaming helps to avoid accidental execution of such files in these operating systems and therefore prevents infection.

`spm` action makes **Dr.Web Scanner** move infected or suspicious files to the quarantine directory (`%var_dir/infected/` by default). This option actually has a little value since infected and suspicious files for other operating systems can not infect or damage UNIX system. Moving of suspicious files for UNIX system may cause system malfunction or failure.

Recommended command for day-to-day scanning:

```
$ drweb <path> -cu -icd -spm -ar -ha -fl- -ml -sd
```

You can save this command to the text file and convert it into simple shell script with command:

```
# chmod a+x [filename]
```

Dr.Web Scanner default settings could be changed in the configuration file.