Dr.WEB®

Anti-virus
for Novell Storage Services

Defend what you create

Administrator Manual

Dr.Web® Anti-virus for Novell Storage Services
Version 6.0.2
Administrator Manual
05.12.2014

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

The manual is designed for a person responsible for anti-virus protection and security ("Administrator" hereinafter).

Dr.Web® Anti-virus 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. Although most malware is aimed at non-UNIX systems, file servers can be used for distribution of viruses for all operating systems including macro-viruses for applications.

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

Dr.Web for Novell Storage Services includes the following components:

- **Dr.Web Scanner** - console anti-virus scanner that provides detection and neutralization of viruses on the local machine and in the shared directories;
- **Dr.Web Daemon** - a background that performs functions of an external anti-virus filter;
- **Dr.Web Monitor** - a resident component that runs and terminates other **Dr.Web** modules in the required order;
- **Dr.Web Agent** - a resident component that helps to configure and manage **Dr.Web** components, gathers statistics and provides integration with **Dr.Web Enterprise Security Suite** (**Dr.Web ESS**);

By default, the solution includes **Dr.Web Agent**, designed for integration with **Dr.Web ESS** 6.0. If you want to integrate the suite with **Dr.Web ESS** 10.0, install the updates for **Dr.Web Agent** and perform additional configuration steps. For details, refer to the Dr.Web Agent section.

- **Dr.Web Engine** and virus databases that are regularly updated;
- **Dr.Web Updater** (implemented as a **Perl** script) - a component that provides regular updates to virus databases;
- **NSS Daemon** – main component that is responsible for integration with *NSS* file system;

The following picture shows the structure of **Dr.Web for Novell Storage Services** and its components.

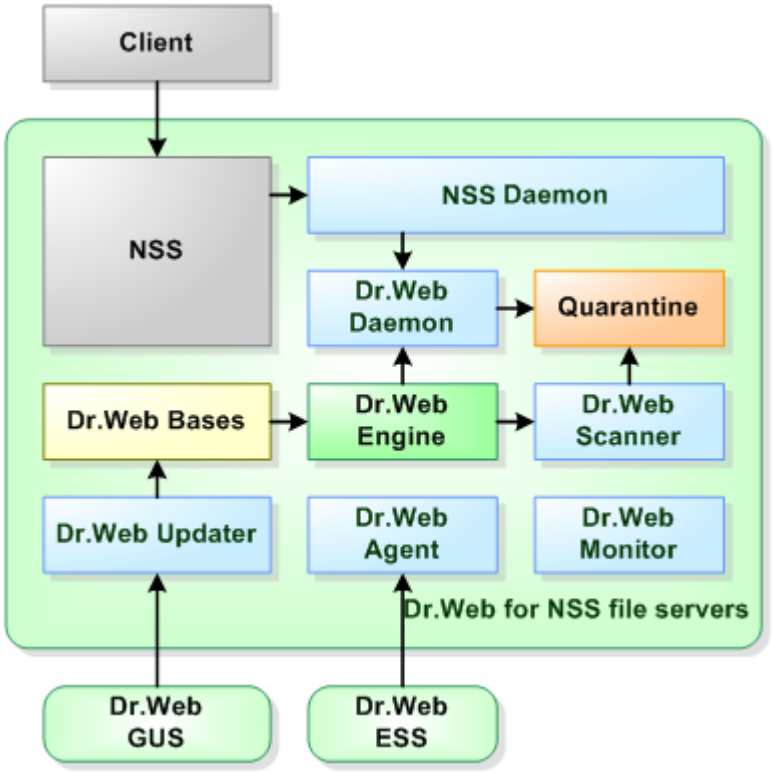


Figure 1. Structure of Dr.Web for Novell Storage Services and its components

The present manual provides information on setup, configuration, and usage of **Dr.Web for Novell Storage Services**, that is:

- General product description
- Installation of **Dr.Web for Novell Storage Services**
- Running **Dr.Web for Novell Storage Services**
- Usage of **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 can find contact information for technical support.

Doctor Web products are constantly developed. Updates to virus databases are issued daily or even several times a day. New product versions appear. They include enhancements to detection methods, as well as to the means of integration with UNIX systems. Moreover, the list of applications compatible with **Doctor Web** is constantly expanding. Therefore, some settings and functions described in this Manual can slightly differ from those in the current program version. For details on updated program features, refer to the documentation delivered with an update.

Terms and Abbreviations

The following conventions are used in the Manual:

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.

Convention	Description
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 a user. For command-line input, it indicates parameter values.
CAPITAL LETTERS	Names of keys and key sequences.
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, where the suite components are installed, the following conventions are used: `%bin_dir`, `%etc_dir` and `%var_dir`:

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

The following conventions are used in the 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
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** operating system (10 SP3, 11 SP1, 11 SP2);
- Installed **Novell Storage Services** (**NSS**);
- *NSS* file system mounted to the specified directory;
- Installed **Perl** 5.8.0 or later for **Dr.Web Updater**.

Hardware requirements are similar to those for selected version of **SUSE Linux Enterprise Server** operating system.

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

Graphic installer of **Dr.Web for Novell Storage Services** requires **X Window System**. To enable the configuration script to run in graphic mode, install **xterm** or **xvt** terminal emulator.

Depending on the range of problems to be solved by **Dr.Web for Novell Storage Services** and operational load, meeting additional hardware requirements can be necessary.

Compatibility with Linux Distributions

Dr.Web for Novell Storage Services is compatible with the following **Linux** distribution – **SUSE Linux Enterprise Server (10 SP3, 11 SP1, 11 SP2)**.

Package File Location

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

- `%bin_dir` – directory with executable modules of **Dr.Web for Novell Storage Services** and **Dr.Web Updater** (perl script `update.pl`);
- `%bin_dir/doc/` – documentation on the product. All documentation is available in both Russian and English languages and represented in `KOI8-R` и `UTF-8` text files.

`%bin_dir/lib/` – directory with various service libraries and supporting files for **Dr.Web for Novell Storage Services** component operation, for example:
  - `ru_scanner.dwl` – file of **Dr.Web Scanner** language resources.
- `%bin_dir/scripts/` – directory with additional scripts, **Dr.Web for Novell Storage Services** autoconfiguration script, migration script for transfer of configuration from older **Dr.Web** versions.
-
- `%etc_dir/` – directory with **Dr.Web for Novell Storage Services** configuration and enable files that manage startup of components operating in daemon mode*
- `%etc_dir/agent/` – directory with additional configuration files for **Dr.Web Agent**;
- `%etc_dir/monitor/` – directory with additional configuration files for **Dr.Web Monitor**;
- `%etc_dir/templates/` – directory where notification templates are located. Notifications are generated and sent to recipients on detection of malicious objects in an email message or if an error occurs during operation of **Dr.Web Daemon** or its modules;
- `%var_dir/bases/` – directory with virus databases (`*.vdb` files);
- `%var_dir/infected/` – **Quarantine** folder that serves for isolation of infected or suspicious

files if the corresponding action is specified in **Dr.Web for Novell Storage Services** settings.

- `%var_dir/lib/` – anti-virus engine implemented as a loadable library (**`drweb32.dll`**).

*) Directory of the `enable` files depends on **Dr.Web for Novell Storage Services** installation method:

- Installation using the **universal package for UNIX systems**:
  Files are stored in the `%etc_dir` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation using the **native DEB packages**:
  Files are stored in the `/etc/defaults` directory and named as follows
  `drwebd`,
  `drweb-monitor`.
- Installation using **native RPM packages**:
  Files are stored in the `/etc/sysconfig` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.

Configuration Files

General format of configuration files

All **Dr.Web for Novell Storage Services** settings are stored in configuration files which you can use to configure all suite components. Configuration files are text files, so they can be edit in any text editor. They have the following format:

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

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

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

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

Configuration files are formed according to the following rules:

- Symbols '`;`' or '`#`' mark the beginning of a comment. Text that follows these symbols is ignored by **Dr.Web for Novell Storage Services** modules when reading a file.
- Contents of the file is divided into sets of named sections. Possible section names are hardcoded and cannot be changed. The section names are specified in square brackets.
- Each file section contains configuration parameters, grouped by meaning.
- One line contains a value (or values) only for one parameter.
- General format for parameter value setting (spaces enclosing the '=' signed are ignored) is the following:

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

- Parameter names are hardcoded and cannot be changed.
- Names of all sections and parameters are case insensitive.

- Order of sections in a file and order of parameters in sections are of no consequence.
- Parameter values in a file may be enclosed in quotation marks (and must be enclosed in quotation marks if they contain spaces).
- Some parameters can have more than one value. In this case, parameter values are separated by a comma or each parameter value is set separately in different lines of the configuration file. If values of a parameter are separated by commas, spaces between a comma and a value are ignored. If a space is a part of a value, the whole value must be enclosed in quotation marks.

If a parameter can have several values, that is explicitly designated. If the possibility to assign several values to a parameter is not explicitly designated, the parameter can have only one value.

Example of assigning several values to a parameter:

1) Separating values by commas:

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

2) Setting of each parameter value separately:

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

If a parameter is not specified in a configuration file, this does not mean that the parameter does not have any value. In this case, the parameter value is assigned by default. Only a few parameters are optional or do not have default values, which is mentioned separately.

Parameter description rules used in this Manual

Each parameter in this manual is described as follows:

ParameterName = {Parameter type | Possible values}	Description {Whether more than one value is possible} {Special remarks} {Important remarks}
	Default value: **ParameterName** = {value | nothing}

Description of parameters is provided in this document in the same order as they are specified in the corresponding configuration file created upon **Dr.Web for Novell Storage Services** installation.

The `Parameter type` field can be one of the following:

- **`numerical value`** — parameter value expressed as a whole non-negative number.
- **`time`** — parameter value expressed as a date unit. The value is a whole number that can be followed by a symbol defining the type of a date unit (`s` – seconds, `m` – minutes, `h` – hours; symbol is case insensitive). If the value does not have a symbol, the parameter is expressed in seconds (by default).

  Examples: `30h`, `15m`, `6` (in the last example, time is expressed in seconds).
- **`size`** — parameter value expressed as a unit of memory size (disk space or RAM). The value is a combination of a whole number that can be followed by a symbol defining the type of a memory size unit (`b` – bytes, `k` – kilobytes, `m` – megabytes, `g` – gigabytes; symbol is case insensitive). If the value does not have a symbol, the parameter is expressed in bytes.

  Examples: `20b`, `15k`

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

  First digit in the value defines permissions for the file owner, second digit - for owner's group, and third digit - for all other users (neither owners nor members of the group).

  Examples: `755`, `644`
- **logical** (**Yes**/**No**) — parameter value expressed as a string that can be one of the following: "`Yes`" or "`No`".
- **path to file/directory** — parameter value expressed as a string which contains a path to a file or folder in the file system. Note, that names of files and folders are case sensitive. If mentioned, you can specify a file mask as a parameter value. A **mask** can include the following symbols:
  - `?` – replaces one symbol in the file (folder) name;
  - `*` – replaces any sequence of symbols (including an empty sequence) in the file (folder) name.

  Example: "`?.e*`" – this mask defines all files with a name consisting of only one character and with an extension which is of any length and starts with "e" (`x.exe`, `g.e`, `f.enable` and others).
- **action** — parameter value expressed as a string which contains actions (those that are applied to objects by **Dr.Web for Novell Storage Services** components). In some cases, the parameter can have one basic and three additional actions specified (in such a case, the name of the parameter type is **actions list**). Basic action must be the first in the list. Different parameters can have a different action list and, in this case, it is specified separately for each parameter. For information on available actions, see Allowed actions.
- **address** — parameter value expressed as a string which contains socket address of a **Dr.Web for Novell Storage Services** component or used external program.
  Address is of the following format: `TYPE:ADDRESS`. There are three available `TYPE`s:
  - `inet` — a TCP socket, `ADDRESS` is specified in the following format: `PORT@HOST_NAME`, where `HOST_NAME` can be either a direct IP address or domain name of the host.

    Example:

    ```
    Address = inet:3003@localhost
    ```

  - `local` — a local UNIX socket, `ADDRESS` is a path to the socket file.

    Example:

    ```
    Address = local:%var_dir/.daemon
    ```

  - `pid` — a real process address that is to be read from the process PID file. This address type is allowed only in certain cases that are explicitly designated in the parameter description.
- **text value, string** — parameter value expressed as a text string. The text can be enclosed in quotation marks (and the text must be enclosed in quotation marks if it contains spaces).
- **log level** — parameter value expressed as a string which contains the verbosity level of logging into the file or **syslog** system service.
- **value** — parameter has the type that is not described in the previous items of the list. In this case, all available values are provided.

Behaviour of the modules if configuration file parameters are ill-defined

- If any parameter value is incorrect, the respective **Dr.Web for Novell Storage Services** module outputs an error message and terminates.

- If any unknown parameter is found when loading a configuration file, **Dr.Web for Novell Storage Services** logs the corresponding message and continues operation in the normal mode.

Some parameters can use regular expressions as values (that is mentioned in the description of the corresponding parameter). Regular expression syntax of **Perl** is used by default. For information on regular expressions, see a corresponding article, for example, on the **Wikipedia** website (Regular expressions article).

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 of 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 the verbosity level, use the **`LogLevel`** parameter. You can also specify additional parameters for certain plug-ins 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 plug-in, 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** component 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 **SyslogFacility** parameter value in configuration files. You can specify one of the following parameter values:

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

Note that if information is logged into **syslog**, an additional parameter - **SyslogPriority** - can be specified in configuration files. **SyslogPriority** 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, **SyslogPriority** is ignored. Otherwise, information is logged into **syslog** with the less verbosity level.

Example:

Let us assume that logging of component operation 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.

Different parameters can have different available actions, they are listed in each parameter description.

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 **NSS Daemon**:

- 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 equals to report).

Installation and Deinstallation

Below you can find detailed description of **Dr.Web for Novell Storage Services** installation, update and uninstallation procedures in UNIX systems. You need superuser (`root`) privileges to perform these operations. To get it, use the **`su`** command or **`sudo`** prefix.

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

You can install, deinstall, and update **Dr.Web for Novell Storage Services** in one of the following ways:

- using GUI;
- using console scripts.

During installation, dependencies are supported, that is if a component installation requires other components to be installed in the system (for example, `drweb-daemon` package requires **`drweb-common`** and **`drweb-bases`** packages), they will be installed automatically.

If you install **Dr.Web for Novell Storage Services** to a computer where other **Dr.Web** products have been previously installed from EPM packages, then at every attempt to remove a module via graphical installer 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 uninstallation 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 in this distribution:

- `drweb-common`: contains the main configuration file - `drweb32.ini`, libraries, documentation and directory structure. During installation of this component, `drweb` user and `drweb` group are created;
- `drweb-bases`: contains Anti-virus search Engine (**Dr.Web Engine**) and virus databases. It requires `drweb-common` package to be installed;
- `drweb-libs`: contains common libraries for all the components of the suite;
- `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 installed;
- `drweb-agent`: contains **Dr.Web Agent** executable files and its documentation. It requires `drweb-common` and `drweb-boost147` packages to be installed;
- `drweb-agent-es`: contains files required for communication between **Dr.Web Agent** and **Dr.Web ESS** server version 6 in central protection mode. It requires `drweb-agent`, `drweb-`

updater and drweb-scanner packages to be installed;

- drweb-agent10: contains executable files and documentation for the updated **Dr.Web Agent** (designed for operation with **Dr.Web ESS** server version 10).
- drweb-agent10-es: contains files required for communication between the updated **Dr.Web Agent** and **Dr.Web ESS** server version 10 in central protection mode.
- 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 installed;
- drweb-monitor: contains **Dr.Web Monitor** executable files and its documentation. It requires drweb-agent, drweb-common and drweb-boost147 packages to be installed;
- drweb-perftools0: contains **Google Performance Tools** library used by **NSS Daemon**. It requires drweb-libs package;
- drwen-nss-doc: contains **Dr.Web for Novell Storage Services** documentation;
- drweb-nss: contains **NSS Daemon** executable files and its documentation. It requires drweb-common, drweb-perftools0, drweb-agent and drweb-monitor packages.

In distributions for 64-bit systems, two additional packages are included: drweb-libs and drweb-libs32, which contain libraries for 64 and 34-bit systems correspondingly.

To install all **Dr.Web for Novell Storage Services** components automatically, 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 is created, and the GUI installer starts. If it starts without root privileges, the GUI installer tries to gain required privileges.

If the GUI installer fails to start, then interactive console installer starts automatically.

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

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

After you extract the content, you can start the GUI installer and continue setup with the following command:

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

To install with the use of the console installer, use the following command:

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

Installation, regardless of the used method, includes the following steps:

- 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 installed to the corresponding directories.
- Other files are installed. If a file with the same name already exists in the directory (e.g. after inaccurate removal of previous package versions), it is overwritten with the new file, and a copy of

the old one is saved as `[file_name].O`. If a file with the `[file_name].O` name already exists in this directory, it is replaced with the new file.

Please note that if the used **Linux** distribution features **SELinux**, installation can be interrupted by the security subsystem. If such situation occurs, set **SELinux** to `(Permissive)` mode. To do this, enter the following command:

```
# setenforce 0
```

and restart the installer.

After the installation completes, configure **SELinux** security policies to enable correct operation of anti-virus components.

You can remove the `drweb-nss_[version]~linux_[processor_architecture]` directory and .run file after successful completion of installation.

Using GUI Installer

To install with GUI

1. Enter the following command:

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

The setup program launches. On the Welcome screen, click **Next**.

At any step you can return to the previous one by clicking **Back**. To continue installation, click **Next**. To abort installation, click **Cancel**.

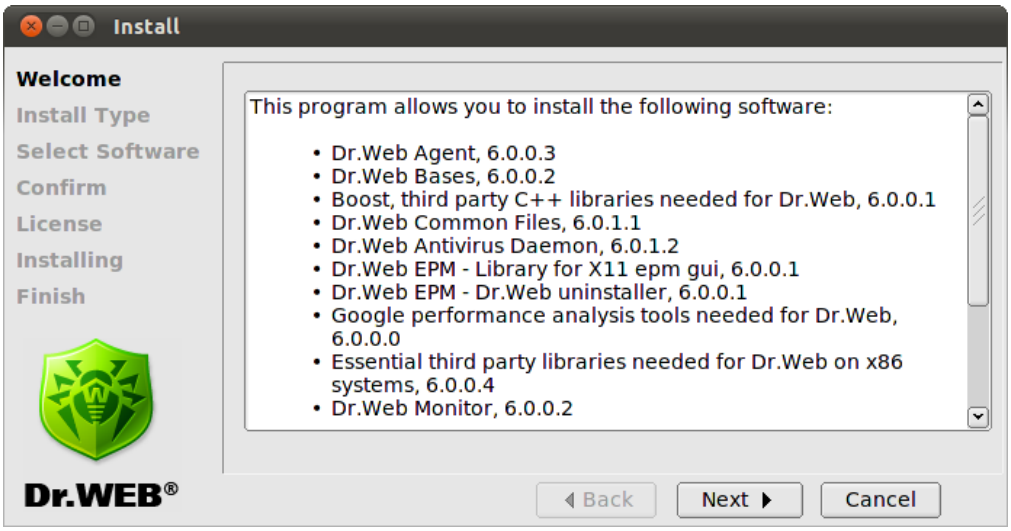


Figure 2. Welcome screen

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

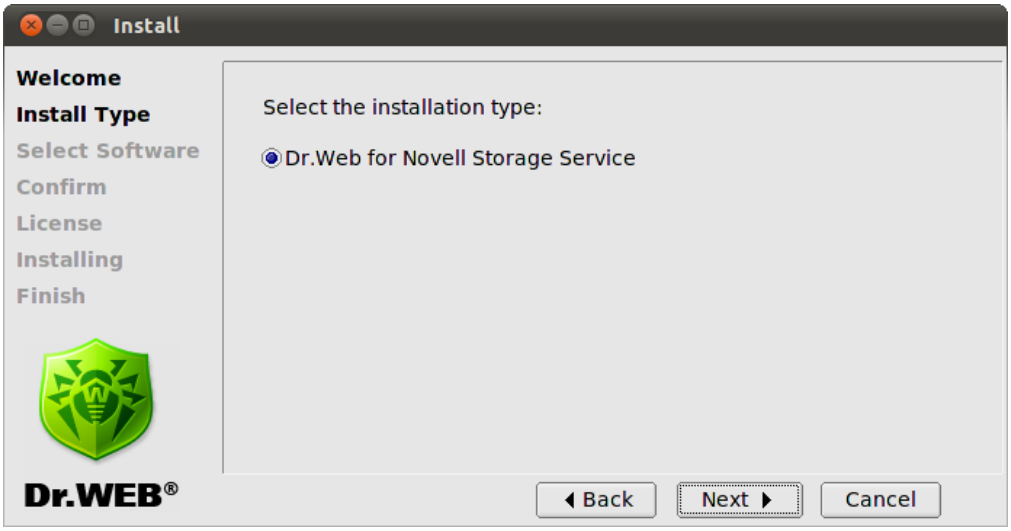


Figure 3. Install type window

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

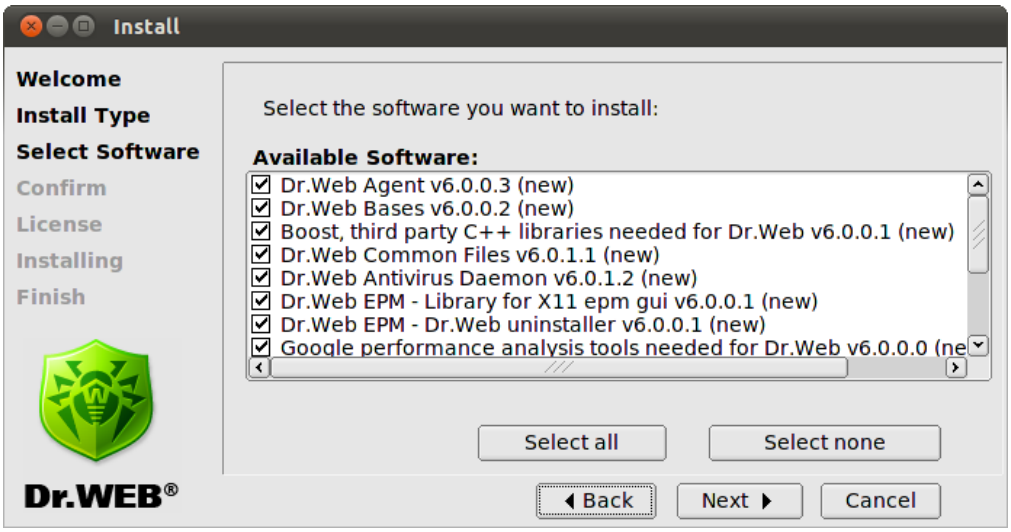


Figure 4. Select Software screen

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 installed automatically.

Click to **Select all** to select all components. Click **Install None** to clear selection.

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

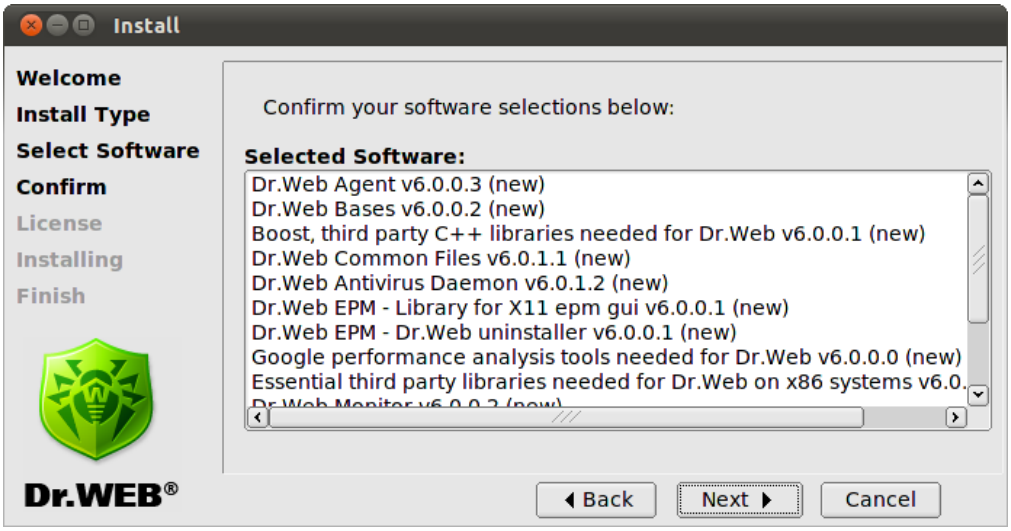


Figure 5. Confirm screen

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 a preferred language of the agreement (Russian and English languages are available):

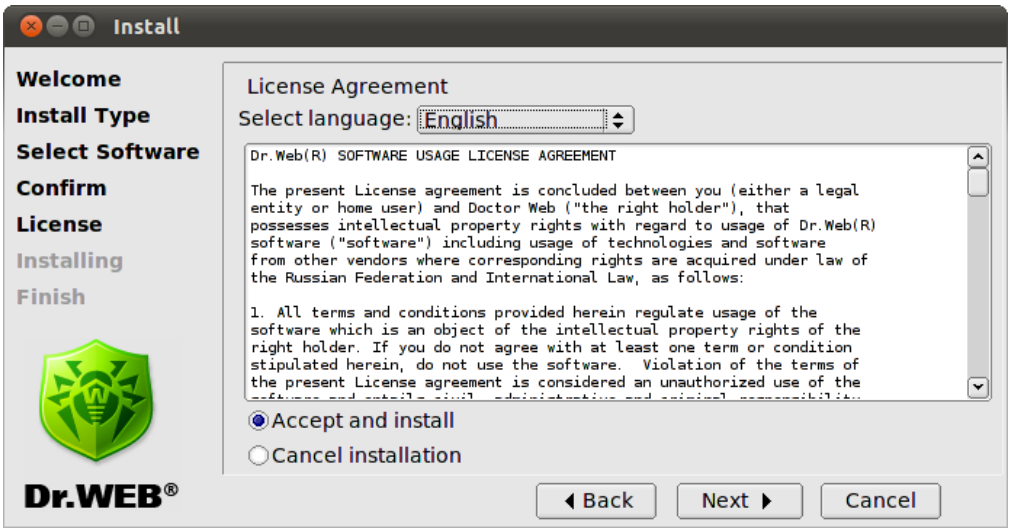


Figure 6. License Agreement screen

5. After you accept the **License Agreement**, installation starts. On the **Installing** screen, you can review the installation process in real-time:

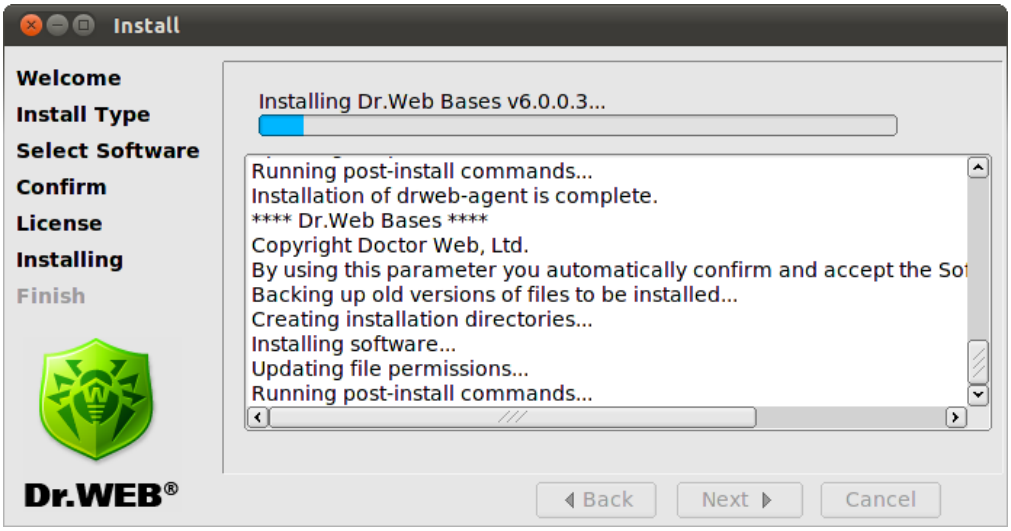


Figure 7. Installing screen

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 **Run interactive post-install script**, once component installation completes, the post-install script for **Dr.Web for Novell Storage Services** basic configuration initializes.

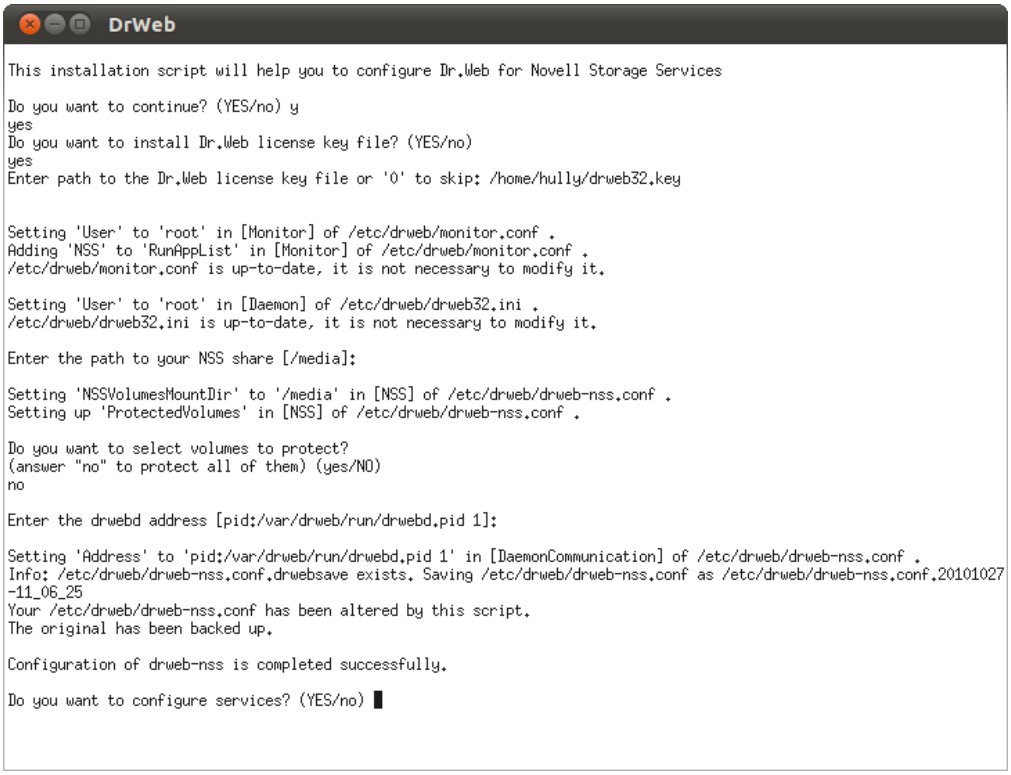


Figure. 8. Interactive post-install script

After initialization of the script, you can:

- install license key file which you received after product registration;
- specify a path to the folder where NSS partitions is mounted (NSS share);
- specify, if required, NSS partitions to be protected from viruses (by default, all partitions are protected);

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

If configuration files already exist, their backup copies with the `.drwebsave` extension are created before the files are modified.

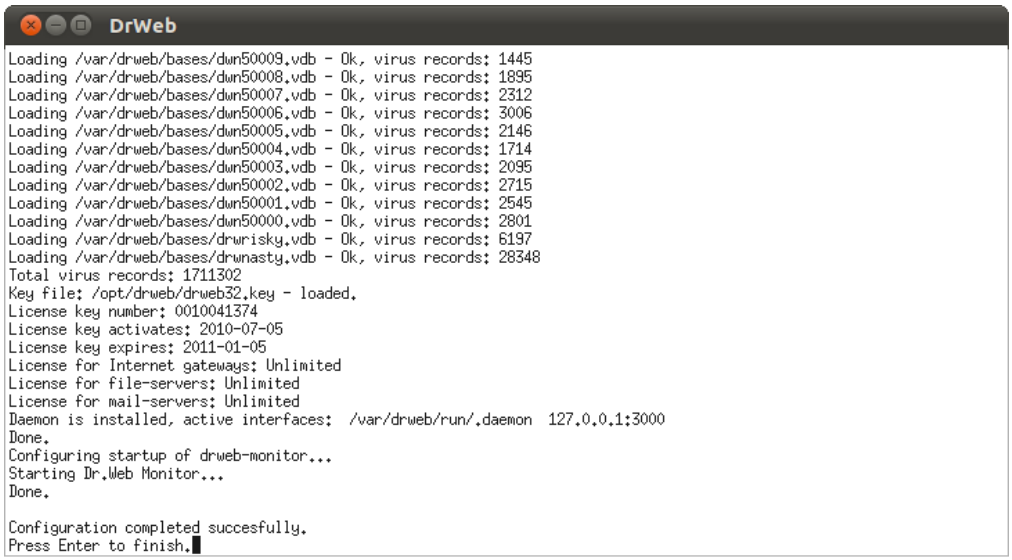


Figure 9. Starting services

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

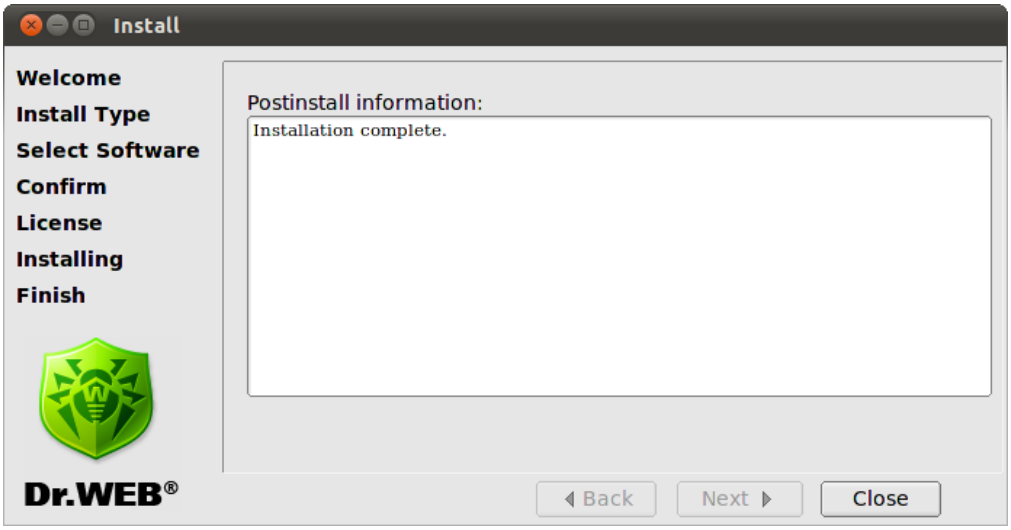


Figure 10. Finish screen

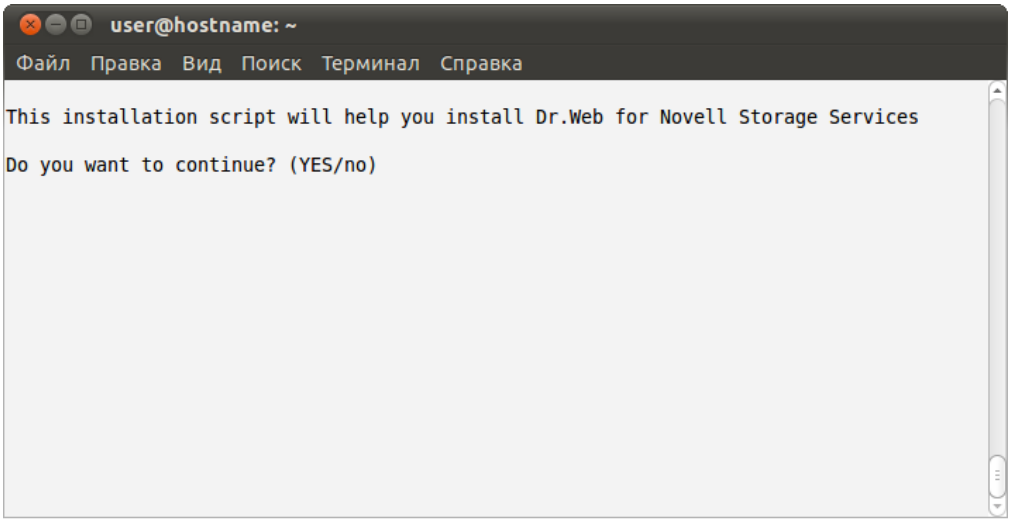
Using Console Installer

Console installer starts automatically if the GUI installer fails to start. If the console installer also fails to start (for example, if it is impossible to gain necessary privileges), you can try to run the following command with `root` privileges:

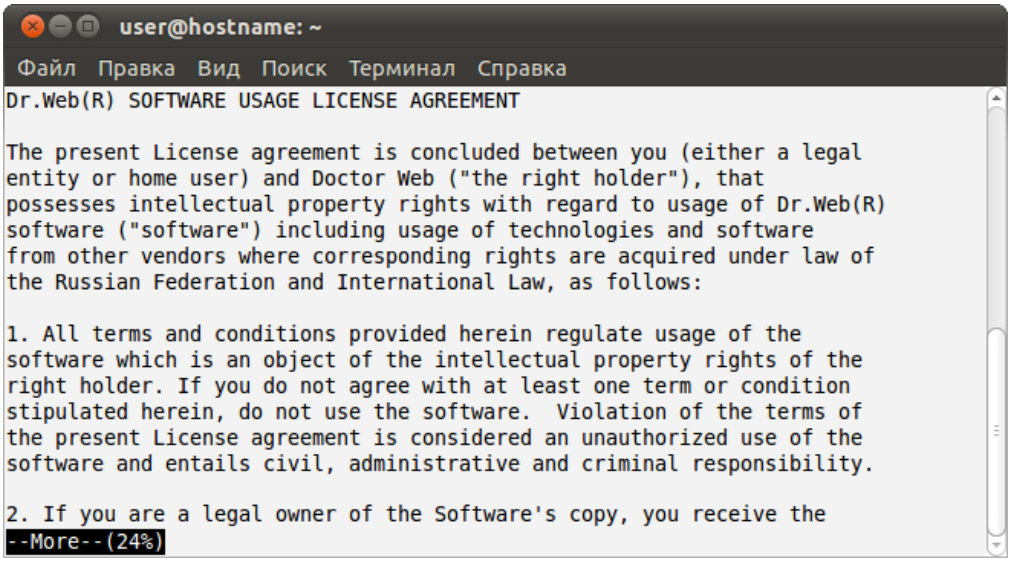
```
# drweb-nss_[version]~linux_[processor_architecture]/setup.sh
```

To install from console

1. Once the console installer starts, the following dialog window opens:

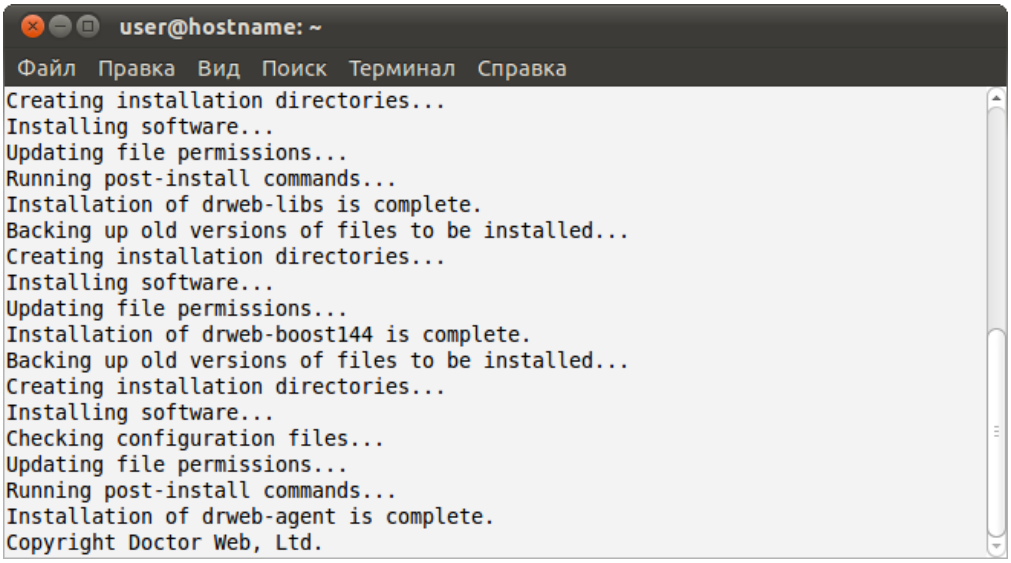


2. If you want to install **Dr.Web for Novell Storage Services**, enter **Y** or **Yes** (values are case insensitive), otherwise enter **N** or **No**. Press ENTER.
3. Review the **License Agreement**. To scroll the text, press SPACEBAR:

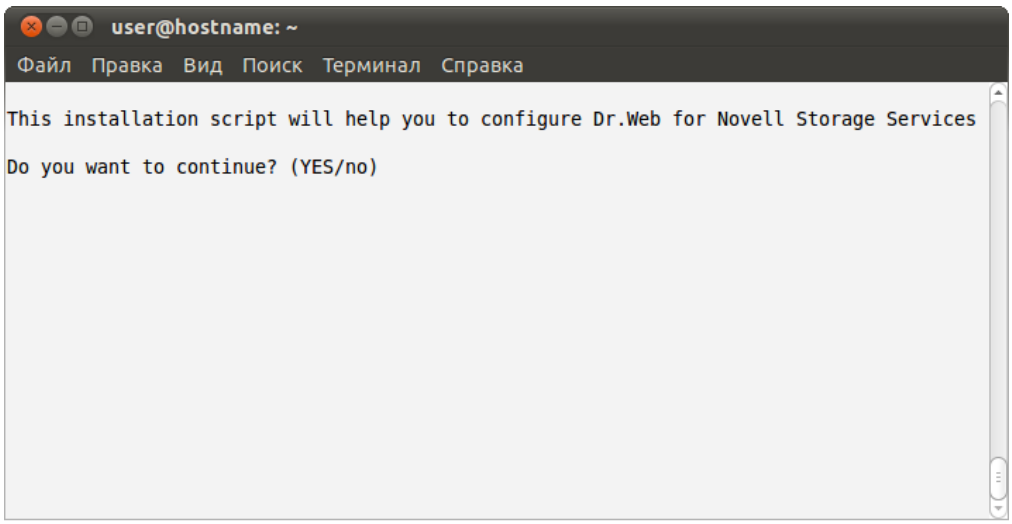


To continue the installation, you need to accept the **License Agreement**. If you agree to the terms, enter **Y** or **Yes**. Otherwise, the installation aborts.

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



5. Once installation of the components completes, the post-install script runs automatically to set up **Dr.Web for Novell Storage Services** basic configuration. You are offered to specify the path to the license key file and automatically enable all the services necessary for **Dr.Web for Novell Storage Services** proper operation (for example, **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**). In addition, you can specify the path to the directory where NSS partitions are mounted and select NSS partitions to be protected from viruses (by default, all partitions are protected).



Removing Distribution Package for UNIX Systems

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

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

If startup is performed without root privileges, the GUI uninstaller tries to gain appropriate privileges.

If the GUI uninstaller fail to start, then interactive console uninstaller is initialized.

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

During uninstallation, the following actions are performed:

- Original configuration files are removed from the `%etc_dir/software/conf/` directory.
- If operational copies of configuration files are not modified by the user, they are also removed. If the user made any changes to them, they are preserved.
- Other **Dr.Web** files are removed. If a copy of an old file was created during installation, this file is restored under the name it had before the installation. Such copies are usually named `[file_name].O`.
- License key files and log files are saved to their corresponding directories.

Using GUI Uninstaller

To uninstall with GUI

1. Enter the following command:

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

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

At any step, you can return to the previous stage by clicking **Back**. To continue installation, click **Next**. To abort uninstallation, click **Cancel**.

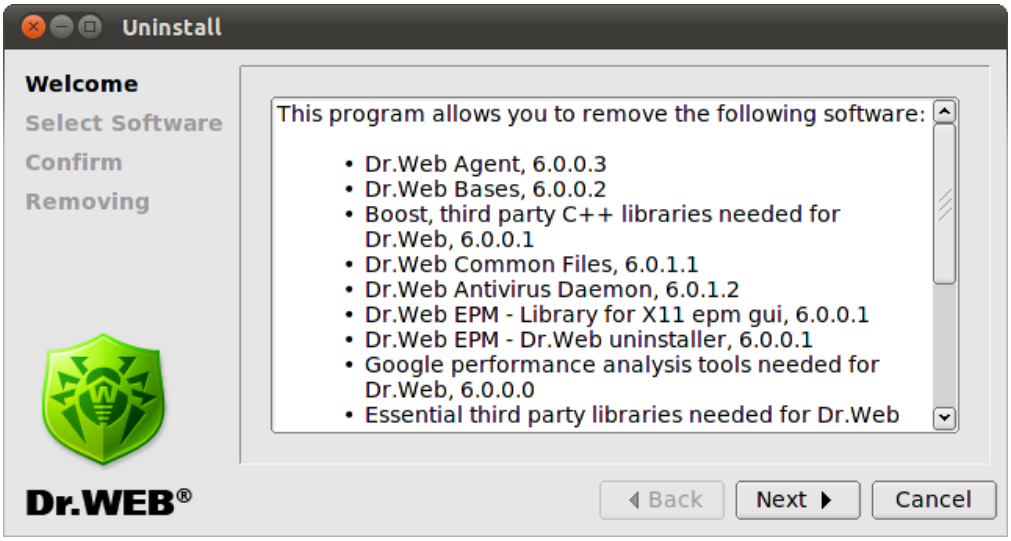


Figure 11. Welcome screen

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

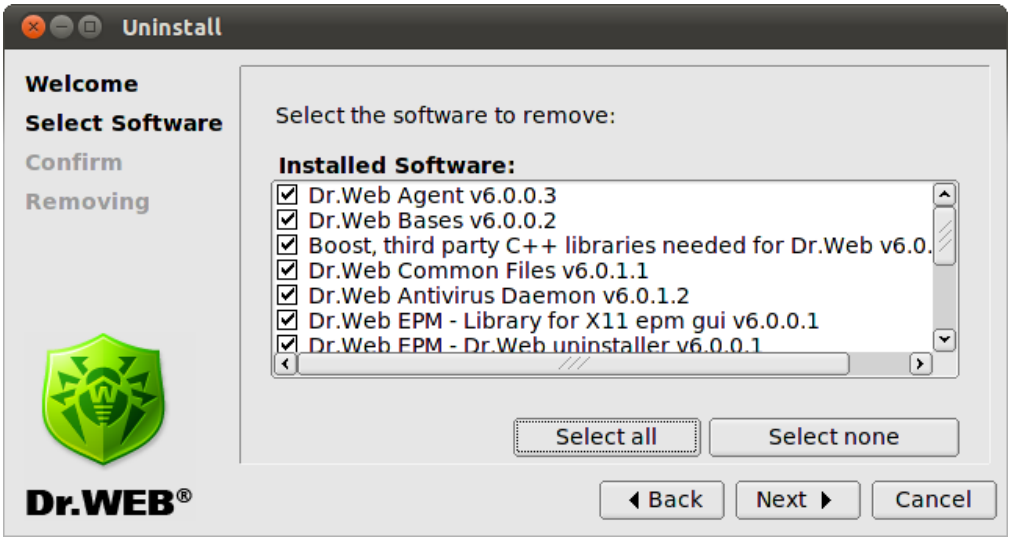


Figure 12. Select Software screen

All corresponding dependencies are selected to be uninstalled automatically.

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

Click **Select All** to select all components. To clear selection, click **Select None**.

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

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

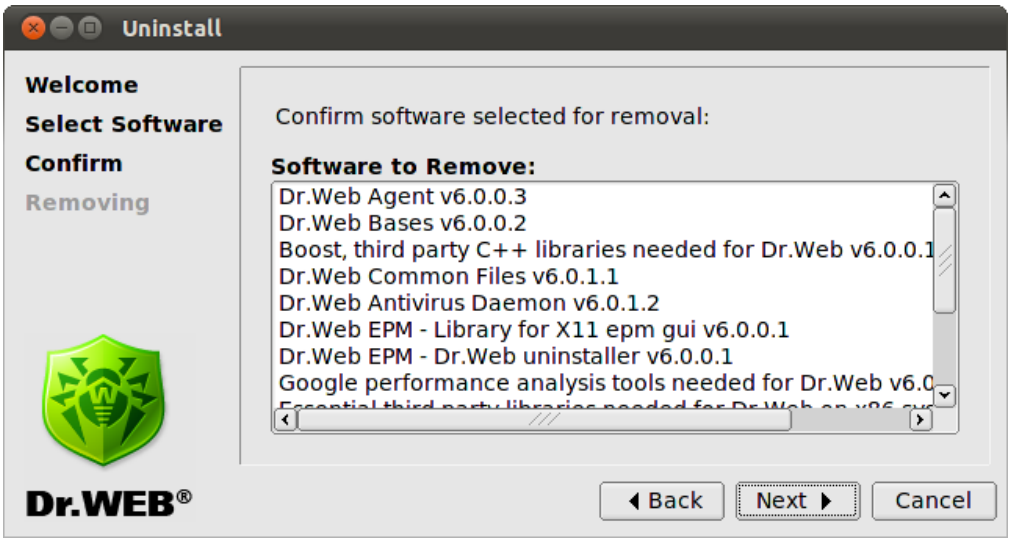


Figure 13. Confirm screen

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

4. On the **Removing** screen, you can review results of the uninstallation steps in real time:

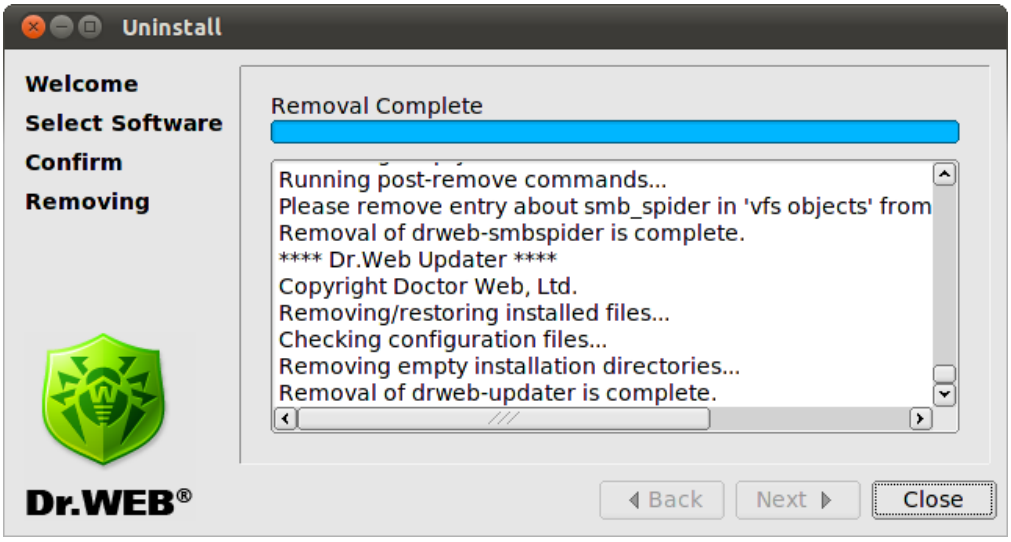


Figure 14. Removing screen

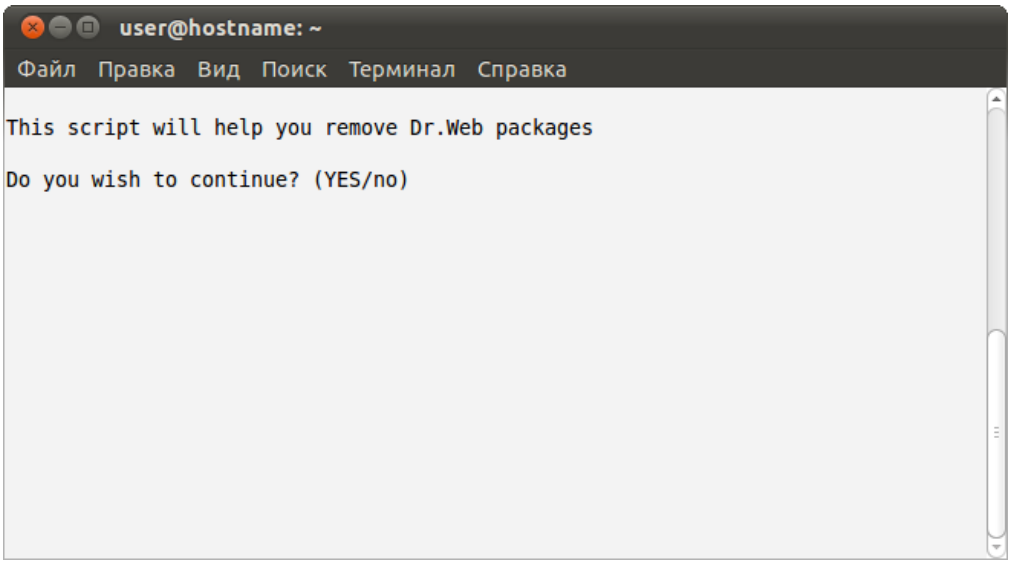
5. Click **Close** to exit setup.

Using Console Uninstaller

Console uninstaller starts automatically when graphical uninstaller fails to start.

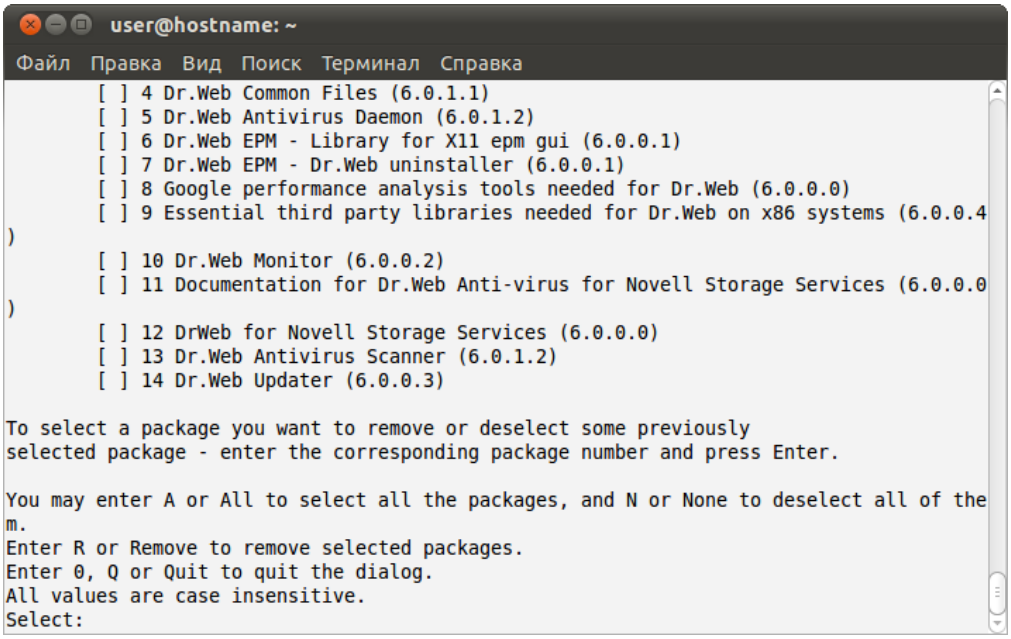
To uninstall from console

1. Once the console uninstaller starts, a dialog window opens:

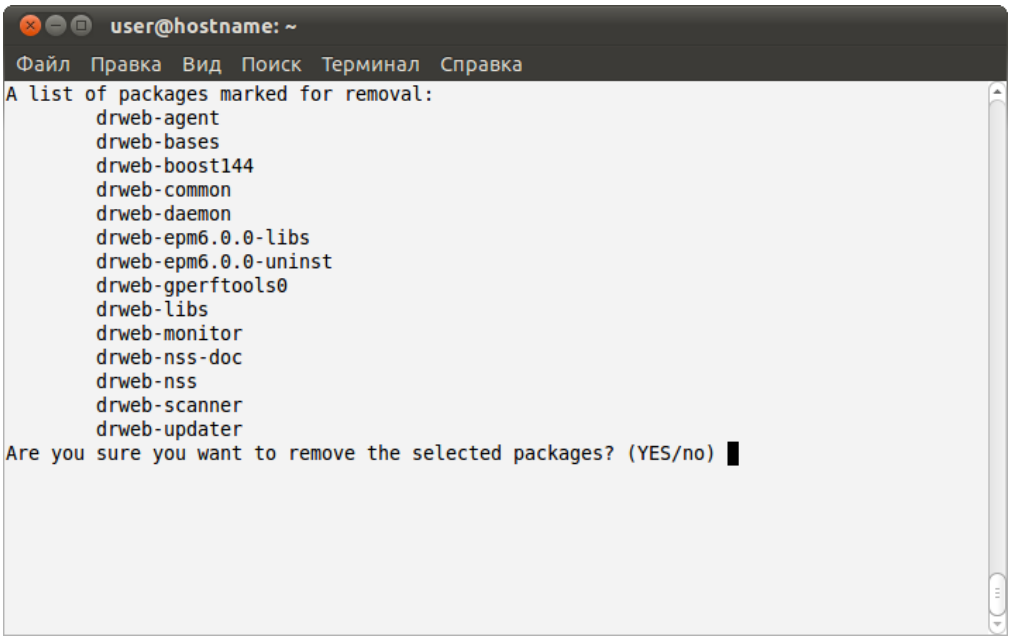


   If you want to uninstall **Dr.Web for Novell Storage Services**, enter **yes**, otherwise enter **no**. Press ENTER.

2. Review the list of components available for removal:



3. To select components to remove, follow the prompts .
4. To confirm you selection and start uninstallation, enter **Y** or **Yes** (they are case insensitive) and press ENTER:



5. You can results of the uninstallation steps in the console in real time.
6. Once the process completes, exit setup.

Updating Distribution Package for UNIX Systems

Updating procedure combines installation and deinstallation procedures. To update **Dr.Web for Novell Storage Services**, download the latest version of the corresponding software, remove the previous version and install the new one.

After an update, license key files, log files, and configuration files modified by the user are remained in the corresponding directories.

Installing from Native Packages

All packages are located in the **Dr.Web** official repository http://officeshield.drweb.com/drweb/. Once you added the repository to the package manager of your system, you can install, update or remove necessary packages like any other program from repository. All dependencies are resolved automatically.

After installing packages from repository, automatic post-install script for installing license key file is not initiated. Licence key file must be manually copied to `%bin_dir`.

For the updates to take effect, you need to restart all **Dr.Web** services after updating from repository.

All the following commands to add repositories, import keys, install and remove packages must be run with administrator privileges (`root`).

If it is necessary, use the **`sudo`** or **`su`** commands.

Zypper package manager (SUSE Linux)

1. Installation:

To add the repository, use the following command:

```
zypper ar -t YUM http://officeshield.drweb.com/drweb/el5/stable/i386/ drweb
```

or

```
zypper ar -t YUM http://officeshield.drweb.com/drweb/el5/stable/x86_64/ drweb
```

To install **Dr.Web for Novell Storage Services**, use the following commands:

```
zypper refresh
zypper install drweb-nss
```

2. Deinstallation:

To remove **Dr.Web for Novell Storage Services**, use the following command:

```
zypper remove drweb-nss
```

To remove all installed packages from **Dr.Web**, use the following command (in some systems, it is required to escape the '`*`' character with a backslash: '`\*`'):

```
zypper remove drweb*
```

Removal with the use of **`zypper`** has the following features:

1. The first variant of the command removes only the `drweb-nss`, package but other packages (which could be automatically installed on the package installation to resolve dependences) remain in the system.
2. The second variant of the command removes from the system all packages, names of which start with the '`drweb`' string (this is a standard pattern for a **Dr.Web** package name). Please note that this command removes from the system all packages which name corresponds to the pattern, not only those of **Dr.Web for Novell Storage Services**.

You can also use alternative package managers (or example, **YaST**) to install or remove the packages.

Starting Dr.Web for Novell Storage Services

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

To run **Dr.Web for Novell Storage Services** manually:

1. Register the software.
2. Copy or move the `drweb32.key` key file to the directory with **Dr.Web** executable files (the default directory is `/opt/drweb/`). Name of the key file can differ in different distribution packages (for details, see Software Registration). To use a key file from another location, specify the full path to it as a **`Key`** parameter value in the `drweb32.ini` main configuration file. Since **Dr.Web for Novell Storage Services** can operate only in the `Standalone` mode (without integration with **Dr.Web Enterprise Security Suite**), path to the key file must be also set as a value of the **`LicenseFile`** parameter in **Dr.Web Agent** configuration file – `agent.conf`.
3. Configure the software by making necessary changes in configuration files. For details on configuration parameters, see the corresponding sections of this Manual.
4. Open `drwebd.enable` file and set the value of the `ENABLE` variable to `1` in order to run **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** on the local machine (properly configured **Dr.Web Daemon** is working on another local network computer), the value of the `ENABLE` variable must be set to `0`.
5. Open `drweb-monitor.enable` file and set the value of the `ENABLE` variable to 1 in order to run **Dr.Web Monitor**.
6. Run **Dr.Web Daemon** and **Dr.Web Monitor** using the command line interface or your file manager. After startup, **Dr.Web Monitor** starts all other **Dr.Web for Novell Storage Services** components. You can also run each module independently, but **Dr.Web Agent** must be run first, since all other modules receive configuration from **Dr.Web Agent**.

Location of the `enable` files depends on **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:
  Files are saved to the `%etc_dir` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation from **native DEB packages**:
  Files are saved to the `/etc/defaults` directory and named as follows
  `drwebd`,
  `drweb-monitor`.
- Installation from **native RPM packages**:
  Files are saved to the `/etc/sysconfig` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.

For Linux and Solaris OS

To run **Dr.Web for Novell Storage Services**:

1. Register the software.

2. Copy or move the key file to the directory with **Dr.Web for Novell Storage Services** executable files (the default directory for UNIX systems is `%bin_dir`). Name of the key file can be different in different distribution packages (for details, see Software Registration):
   - If **Dr.Web for Novell Storage Services** was purchased as a standalone product, license key file is named `drweb32.key`. In this case, copy the file to the `%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**, archive received during registration contains a key file for **Dr.Web 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.

   To use a key file from a different location or with another name (for example, `agent.key`), specify its full path as a `Key` parameter value in the `drweb32.ini` configuration file. In the `Standalone` mode, alternative path to the key file must be specified as a value of the **`LicenseFile`** parameter in `agent.conf` (a configuration file of **Dr.Web Agent**).
3. Configure the software by making necessary changes to the configuration files. For details on configuration parameters, see the corresponding sections of this Manual.
4. Set `1` as a value of the `ENABLE` variable in the `drwebd.enable` file to run **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** on the local machine (properly configured **Dr.Web Daemon** is working on another local network computer), the value of the ENABLE variable must be 0 (its default value).
5. Set `1` as a value of the `ENABLE` variable in the `drweb-monitor.enable` file to run **Dr.Web Monitor**.

Location of the `enable` files depends on **Dr.Web for Novell Storage Services** installation type:

- Installation from **universal package for UNIX systems**:
  Files are saved to the `%etc_dir` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation from **native DEB packages**:
  Files are saved to the `/etc/defaults` directory and named as follows
  `drwebd`,
  `drweb-monitor`.
- Installation from **native RPM packages**:
  Files are saved to the `/etc/sysconfig` directory and named as follows
  `drwebd.enable`,
  `drweb-monitor.enable`.

---

6. Run **Dr.Web Daemon** and **Dr.Web Monitor** either from the console or a file manager of your operation system. After startup, **Dr.Web Monitor** starts all other **Dr.Web for Novell Storage Services** components.

In case of installation from native packages in Solaris:

During **Dr.Web for Novell Storage Services** installation, the SMF service management system attempts to run **Dr.Web Monitor**. If **Dr.Web Monitor** cannot find a licence key file (for example, on the first installation of **Dr.Web for Novell Storage Services**), it stops its operation and SMF goes into the maintenance state.

To run **Dr.Web Monitor**, reset the maintenance state:

- Enter the following command

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

  where FMRI is a unique identifier of a controlled resource. In this case, a unique identifier

of **Dr.Web Monitor** is required.

- Force termination of the process from `svcs -p` output list.

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

where PID is a number of the process listed above.

- Restart **Dr.Web Monitor** with the following command:

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

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

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

To stop the service:

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

The `drwebd` module can be launched in one of the following two modes:

1. with the `init` script (standard launch)
2. with the **Dr.Web Monitor**

In the second mode, set the `ENABLE` parameter to 0 in the `enable` file.

Each of the components can be run independently as well, but note that **Dr.Web Agent** must be started first since all other modules receive configuration from **Dr.Web Agent**.

For FreeBSD OS

To run **Dr.Web for Novell Storage Services**:

1. Register the software.
2. Copy or move the key file (with the `.key` extension) to the directory with **Dr.Web for Novell Storage Services** executable files (the default directory for UNIX systems is `%bin_dir`). Name of the key file can differ in different distribution packages (for details, see Software Registration):
   - If **Dr.Web for Novell Storage Services** was purchased as a standalone product, license key file is named `drweb32.key`. In this case, copy the file to the `%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**, archive received during registration contains a key file for **Dr.Web Enterprise Server** (`enterprise.key`) and a key file for workstations (`agent.key`). Rename `agent.key` to `drweb32.key` and copy the file to the `%bin_dir` directory.

   To use a key file from a different location or with another name (for example, `agent.key`), specify its full path as a **`Key`** parameter value in the `drweb32.ini` configuration file. In the `Standalone` mode, alternative path to the key file must be specified as a value of the **`LicenseFile`** parameter in `agent.conf` (a configuration file of **Dr.Web Agent**).
3. Configure the software by making necessary changes to the configuration files. For details on configuration parameters, see the corresponding sections of this Manual.
4. Add the following lines to the `/etc/rc.conf` file:
   - `drwebd_enable="YES"` – to run **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** on the local machine (properly configured **Dr.Web Daemon** is working on another local network computer), then you do not need to add the line to the `rc.conf` file;

- `drweb_monitor_enable="YES"` – to run **Dr.Web Monitor.**

5. Run **Dr.Web Daemon** and **Dr.Web Monitor** either from the console or from a file manager of your operation system. After startup, **Dr.Web Monitor** starts all other **Dr.Web for Novell Storage Services** components.

Each of the components can be run independently as well, but note that **Dr.Web Agent** must be started first since all other modules receive their configuration from **Dr.Web Agent**.

Configuring SeLinux Security Policies

If the used **Linux** distribution features **SELinux** security subsystem (`Security-Enhanced Linux`), you need to configure security policies used by **SELinux** in order to enable correct operation of anti-virus components (**Dr.Web Daemon** and **Dr.Web Console Scanner**) after the installation.

Moreover, if **SELinux** is enabled, product installation from distribution packages (`.run`) can fail because an attempt to create `drweb` user, whose privileges are used by **Dr.Web for Novell Storage Services**, will be blocked.

Thus, before installing the product, check **SELinux** operation mode with the use of **`getenforce`** command. This command outputs the current operation mode which can be one of the following:

- **`Permissive`** – protection is active, but permissions are supported: actions that violate the security are not denied but logged.
- **`Enforced`** – protection is active and restrictions are enforced: actions that violate the security are logged and blocked.
- **`Disabled`** – **SELinux** is installed but not active.

If **SELinux** is operating in the `Enforced` mode, temporarily (until the product is installed and security policies are configured) enable `Permissive` mode. To do this, enter the **`setenforce 0`** command that temporarily (until the next restart) sets **SELinux** operation mode to `Permissive`. To enable the `Enforced` mode again, enter the **`setenforce 1`** command.

Note that regardless of the mode enabled with the **`setenforce`** command, after system restart **SELinux** will operate in the mode specified in the settings (normally, **SELinux** configuration file is located in the `/etc/selinux` directory).

In general, if **`audit`** daemon is used, the log file resides in `/var/log/audit/audit.log`. Otherwise, notifications on forbidden actions are logged to the following log file: `/var/log/messages`.

For correct operation of anti-virus components when **SELinux** is enabled, compile special security policies once the product installation completes.

Please note that some Linux distributions may not have the below mentioned utilities installed by default. In this case you need to additionally install the required packages.

To create required policies:

1. Create a new file with **SELinux** policy source code (.te file). The file defines restrictions applied to the described module. The source file can be created in one of the two ways:
   1) **With the use of `audit2allow`** utility. This way is more simple. The utility generates permissive rules based on the messages on denial of access to system log files. You can set automatic search of messages in log files or set path to the log file manually.

audit2allow utility resides in the `policycoreutils-python` package, or `policycoreutils-devel` package (for **RedHat Enterprise Linux, CentOS, Fedora** OS, depending on the version), or `python-sepolgen` package (for **Debian, Ubuntu** OS).

Example usage:

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

OR

```
# cat /var/log/audit/audit.log | audit2allow -M drweb
```

In this example, **audit2allow** utility searches for access denied messages in the `audit.log` file.

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

In this example, **audit2allow** searches for access denied messages in log files automatically.

In both cases two files are created as a result of the utility operation: `drweb.te` policy source file and `drweb.pp` policy module which is ready for installation.

In most cases you do not need to adjust policies created by the utility. So, it is recommended to go to step 4 for installation of the `drweb.pp` policy module. Note that **audit2allow** utility outputs **semodule** command invocation string. Copy the string to the command line and execute. That way, you will do instructions of step 4. Go to step 2 only if you want to adjust the policies which are automatically formed for **Dr.Web for Novell Storage Services** components.

2) **With the use of** **policygentool** utility. As a parameter, specify the name of the module which operation you want to configure and the path to its executable file.

Note that **policygentool** utility included in `selinux-policy` package for **RedHat Enterprise Linux** and **CentOS Linux** OS might not function correctly. In this case, use **audit2allow** utility.

Example of creating policies with policygentool:

- For **Dr.Web Console Scanner**:

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

- For **Dr.Web Daemon**:

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

You will be prompted to get information on some domain features and then for each of the modules, 3 files will be created which determine the policy:

`[module_name].te`, `[module_name].fc` и `[module_name].if`.

2. If necessary, edit generated source file of the `[module_name].te` policy and then use the **checkmodule** utility to create a binary representation (`.mod`) of the policy source file.

Please note that for successful policy compilation, a `checkpolicy` package must be installed in the system.

Usage example:

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

3. Create a policy module (`drweb.pp`) with the use of **`semodule_package`** utility.

   Example:

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

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

   Example:

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

After system restart, **SELinux** security subsystem will be configured to enable correct operation of **Dr.Web for Novell Storage Services**.

For details on how to configure **SELinux** and on its operation features, refer to documentation for the used **Linux** distribution.

Registration Procedure

Permissions to use **Dr.Web for Novell Storage Services** are specified in the key file.

License key file contains the following information:

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

By default, the license key file is located in the directory with **Dr.Web for Novell Storage Services** executables.

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

Users who have purchased **Dr.Web for Novell Storage Services** from **Doctor Web** certified partners obtain the license key file. Key files contain the following information which depends on the license type. The license key file also contains information on the user and seller of the product.

For evaluation purposes users may also obtain a demo key file. It allows them 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 can be supplied as:

- a `drweb32.key` file license key for workstations, or as a zip archive containing a 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 **Dr.Web Enterprise 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 can be received in one of the following ways:

- by email as a ZIP-archive containing license key file with `*.key` extension (usually after registration on the website). Extract the license key file using an appropriate archiving utility and copy (or move) it to the directory with **Dr.Web for Novell Storage Services** executable files (default directory for UNIX systems is `%bin_dir`);
- within the distribution package;
- on a separate data carrier as a file with `*.key` extension. In this case, a user must copy it manually to the `%bin_dir` directory.

License key file is sent to a user via email usually after registration on the website (website location is specified in the registration card supplied with the product). Visit the website, fill in the web form with your customer data and submit your registration serial number (printed on the registration card). After that, your license is activated and a key file is created according to the specified serial number. The key file is sent to the specified email address.

It is recommended to keep the license key file until it expires, and use it to reinstall or restore **Dr.Web for Novell Storage Services**. 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 that you provided during the registration; only the email address can be changed (in this case, a license key file will be sent to the new email address). If the serial number matches any entry in **Dr.Web for Novell Storage Services** database, the corresponding key file will be automatically dispatched to the specified email address.

One serial number can be registered no more than 25 times. If you need to recover a lost license key file after its 25th registration, send a request for license key file recovery at http://support.drweb.com/

request/ stating the data input during registration, valid email address, and detailed description of your problem. The request will be considered by **Dr.Web for Novell Storage Services** technical support service engineers. If the request is approved, a license key file will be provided via automatic support system or dispatched via email.

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

Example:

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

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

If the license expires in less than two weeks, **Dr.Web Scanner** outputs a warning message on its startup and **Dr.Web Daemon** notifies the user via email. Messages are sent on every startup, restart or reload of **Dr.Web Daemon** for every license key file installed. To enable this option, set up the **MailCommand** parameter in the [Daemon] section of the drweb32.ini configuration file.

If you want to use a key file from another location, specify the full path to it as a **LicenseFile** parameter value in the [StandaloneMode] section of the **Dr.Web Agent** configuration file (see [StandaloneMode] section description).

Dr.Web for Novell Storage Services

Interacting Modules

Dr.Web for Novell Storage Services provides anti-virus protection for NSS file system using the following interacting modules:

- **NSS Daemon** – resident module used for integration with NSS file system
- **Dr.Web Daemon** – resident module used for checking files for viruses and other threats
- **Dr.Web Monitor** – utility module used for starting, restarting, and terminating **Dr.Web** modules in the specified order and monitoring their operation
- **Dr.Web Agent** - module that allows integration with **Dr.Web Enterprise Security Suite** and gathers statistics on module operation.

Operation principle

NSS Daemon monitors selected NSS volumes and processes modified files according to the settings. You can specify NSS volumes to be monitored in the `[NSS]` section in the `drweb-nss.conf` configuration file:

- if the **`ProtectedVolumes`** parameter value is set, **NSS Daemon** monitors the volumes listed in this parameter;
- if the **`ProtectedVolumes`** parameter value is not set, **NSS Daemon** monitors all volumes mounted in the directory listed in the **`NSSVolumesMountDir`** parameter value.

Before files are sent for scanning, they are prefiltered. Thus, those that satisfy at least one of the following criteria are not scanned:

- zero file size
- file size is greater than the **`MaxFileSizeToScan`** parameter value in the `[NSS]` section (only if that value is not zero)
- file path is both specified as the **`ExcludedPaths`** parameter value in the `[NSS]` section and NOT specified as the **`IncludedPaths`** parameter value.

Files that do not satisfy the criteria mentioned above are added to the internal queue for scanning. Upon the receipt of `SIGHUP` signal, **NSS Daemon** outputs the list of queued tasks if the logging verbosity level is set to `INFO`. Scanning tasks are processed by the thread pool which can be configured with the **`CheckPoolOptions`** parameter in the `[NSS]` section: for example, enable gathering of internal statistics on **NSS Daemon** thread pool.

Files that must be scanned are sent to **Dr.Web Daemon**. You can configure interaction with **Dr.Web Daemon** in the `[DaemonCommunication]` section. **NSS Daemon** can simultaneously operate with **Dr.Web Daemon** running on the local machine and with **Dr.Web Daemons** running on remote machines. In the latter case, the components communicate via sockets. You can specify socket addresses and their weights in the **`Address`** parameter in the `[DaemonCommunication]` configuration file section. Weights are used to distribute load on the socket when **NSS Daemon** operates with several **Dr.Web Daemons**: addresses with higher weights receive more scanning requests.

On threat detection, **Dr.Web Daemon** processes files according to the settings specified for the threat type in the `[Actions]` section: for example, removes an object that can compromise the system security, moves the object to **Quarantine** (you can configure **Quarantine** settings in the `[Quarantine]` section). When a threat is detected, notifications can be sent (you can configure notification settings in the `[Notifications]` section). Information on file processing is logged (you can configure logging in the `[Logging]` section).

Moreover, statistics on processed files is sent to **Dr.Web Agent**. You can configure statistics gathering in the `[Stat]` section. Information on a threat is sent immediately after it was detected; general statidstics is sent at intervals specified in the **`SendPeriod`** parameter.

If an error occurs during processing of a file, **NSS Daemon** applies a certain action to it; the action must be specified in the **`ProcessingError`** parameter in the `[Actions]` section.

Command Line Parameters

As any UNIX program, **NSS Daemon** supports command line parameters. You can use the following command to run **NSS Daemon**:

```
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 from **Dr.Web Agent**.

In the current version, **Dr.Web for Novell Storage Services** supports 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 **NSS Daemon** version on the screen and exit		
`-l`	`--level`	`<level>`
Description: Verbosity level for logging information (default value is `info`)		
`-t`	`--timeout`	`<value in seconds>`
Description: Maximum wait time for receiving configuration from **Dr.Web Agent**		
	`--component`	`<name>`
Description: Set the name to be used in requests to **Dr.Web Agent** for configuration information		
	`--log-name`	`<name>`
Description: Component name under which it performs logging		
	`--check-only`	
Description: Start the component in the configuration check mode. To provide correct operation, **Dr.Web Agent** must be previously started. If the configuration test is successful, the following message is output to the console: `Options OK`. If the configuration test failed, the following message is output: `Options ERROR`.		

Example:

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

This command starts the **NSS Daemon** component with 30 seconds time-out for receiving configuration from **Dr.Web Agent** via the `local:/var/drweb/ipc/.agent` socket.

Signals

All resident modules of **Dr.Web for Novell Storage Services** can process the following signals:

- `SIGHUP` – forces modules to reread their configuration files. Upon receipt of this signal by **Dr.Web Monitor**, all modules reread their configuration.
- `SIGINT` and `SIGTERM` – upon receipt of either signal, modules terminate their operation.

NSS Daemon can process the following additional signals:

- `SIGUSR1` – upon receipt of this signal, if the option is enabled, **NSS Daemon** saves files with internal statistics on thread pool and persistent connections to the directory specified as the **`BaseDir`** parameter value in the `[General]` section (for details, refer to the Internal Statistics section).
- `SIGALRM` – upon receipt of this signal, **NSS Daemon** sends all gathered statistics to **Dr.Web Agent**.

Adjustment and Startup

Dr.Web for Novell Storage Services can be started after it is installed with default settings, but to ensure optimal performance, you may adjust it according to your specific requirements.

All **Dr.Web for Novell Storage Services** settings are specified in three configuration files that reside in the `%etc_dir` directory. In the `drweb-nss.conf` configuration file, general **NSS Daemon** settings are specified, in the `agent.conf` file - **Dr.Web Agent** settings, and in the `monitor.conf` file - **Dr.Web Monitor** settings.

Basic **Dr.Web for Novell Storage Services** configuration (providing that all files of the software reside in their default directory) can be performed via the **`configure.pl`** script located in the `%bin_dir/scripts/` directory. After the script is started, it requests values for its main parameters and writes them in the `drweb-nss.conf` configuration file.

The other parameters, required for interaction with NSS file system, are adjusted manually in the `drweb-nss.conf` configuration file.

Checking Configuration

You can validate configuration files and configuration parameters received from **Dr.Web Agent**. For that purpose, use the `--check-only` command line parameter when **Dr.Web Agent**, sending the configuration, is running.

If validation is successful, the following message is output to the console:

```
Options OK
```

If an error is found, the following message with error description is output to the console

```
Options ERROR
```

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

Logging NSS Daemon Operation

Dr.Web for Novell Storage Services can log opeartion using **syslog** system service or saving the information to a log file. In the former case, messages are logged in the following format:

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

where:

- `tid` – identifier of thread which sent the logging message
- `name` – name of the logging module
- `sub` – name of the logging service. The most important services are:
  - `ipc` – service of interprocess communication
  - `thrN` – service supporting the thread pool with the N number
- `level` – log verbosity level. You can specify one of the levels: `FATAL`, `ERROR`, `WARN`, `INFO`, `DEBUG`
- `text` – text of the logged message.

By default, the syslog level is set to `INFO` on startup of a module. After receiving configuration from **Dr.Web Agent**, the verbosity level is adjusted to the specified value.

If it is required to set `DEBUG` level once the module starts (for example, to log information on parameters received from **Dr.Web Agent**), use `--level` command line parameter.

Please note that after configuration is received from **Dr.Web Agent**, the verbosity level is adjusted to the one specified in the configuration file, regardless of the value set with the `--level` parameter.

Statistics

During **Dr.Web for Novell Storage Services** operation, statistics of the following two types is gathered:

1. Internal statistics (statistics on thread pools and connections, which can be used to evaluate load on the software)
2. Statistics on processed files and detected threats.

Internal Statistics

Internal Statistics Files

Upon receipt of `SIGUSR1` signal, the following files of internal statistics are generated:

- `drwebd_client_server_sessionN.txt` – statistics on connections to **Dr.Web Daemon** addresses with the specified weight greater than 0
- `drewbd_client_backup_sessionN.txt` – statistics on connections to backup **Dr.Web Daemon** addresses with the specified weight equal to 0
- `nss_thr.txt` – statistics on **NSS Daemon** thread pool,

where `N` is the address ordinal number in the list specified as **`Address`** parameter value in the `[DaemonCommunication]` section, starting from 0. Addresses with the weight greater than 0 and backup addresses are numbered independently (that is, `drwebd_client_server_session.txt` and `drewbd_client_backup_session` files are allowed to have the same number). At that, if 0 weight is not specified for any address, `drwebd_client_server_sessionN.txt` files are not created. Similarly, if weight greater than 0 is not specified for any address, `drwebd_client_server_sessionN.txt` files are not created.

Statistics on a 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] configuration file section) by specifying an additional parameter **stat** = yes.

Example:

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

Note than when statistics is saved, files are not overwritten; that is, if a file already exists, new data is added to the end of this file.

Format of statistics records

Each statistics record starts with the following lines:

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

where start time of statistics gathering, time of saving statistics to the file, and period for which statistics is gathered are displayed.

For drwebd_client_[server|backup]_sessionN.txt files that contain statistics on connections, a record has the following format:

```
closed: 32 (0.0798005 num/sec)
total created = 34 (0.084788 num/sec)
created on request = 2 (0.00498753 num/sec)
closed by timeout = 0 (0 num/sec)
avg number = 2.58824
max cre = 4 est=3 don=0 act=3
current = 4
```

where:

- closed – number of connections closed for the period (the number is increased each time the statistics is saved)
- total created – total number of established connections
- created on request – number of connections created on request
- closed by timeout – number of connections closed due to timeout
- avg number – average number of unclosed connections
- max – maximum number of
  - cre – created connections
  - est – established but not used connections
  - don – not established connections
  - act – active connections
- current – total number of connections
- num/sec – frequency; that is, parameter value divided by the period length (in seconds)

For the nss_thr.txt file, statistics record is 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
```

where:

- first line contains information on the maximum and minimum number of threads in a pool, type of the pool, maximum time (in seconds) for an additional thread to close upon inactivity;
- second line contains information on the maximum and average number of busy threads;
- third line contains information on the number and frequency of requests to create additional threads;
- fourth line contains information on the number of failed attempts to create threads (such failure can be caused by insufficient resources);
- fifth line contains information on the maximum and average time of processing the requests, in milliseconds;
- sixth line contains information on the current number of threads in a pool and number of busy threads.

Statistics on Processed Files

During **NSS Daemon** operation, statistics of the two following types can be gathered: general statistics and statistics on detected threats.

- **General statistics** is general information on **Dr.Web for Novell Storage Services** operation for a specified period, such as, number of scanned files, their size, number of infected files, etc.
- **Statistics on detected threats** is information on certain files which can compromise system security, for example, files infected with a virus.

General information is gathered in the internal cache and after a certain period (5 minutes, by default) sent to **Dr.Web Agent**. You can adjust the period in the **SendPeriod** parameter of the [Stat] section.

If **NSS Daemon** terminates abnormally, general statistics for this period (between the last time statistics was sent and the component restart) is lost.

Statistics on infected files is sent to **Dr.Web Agent** once a threat is detected.

You can enable or disable statistics gathering by adjusting the **SendToAgent** parameter value in the [Stat] section.

Quarantine

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

When a 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 the original location, permissions, etc.) is created. Its name is the same as the modified name of the quarantined file with an added -info postfix. Permissions for both files are set according to **FilesMode** parameter in the [Quarantine] section.

Example:

eicar.com – original file name;

eicar.comf8JRCG – modified file name;

eicar.comf8JRCG-info – auxiliary file name.

Some additional file properties supported by **NSS** (such as quotas and NSS attributes) can be saved to **Quarantine** together with the file. These properties are automatically reset when the file is restored from the **Quarantine** directory. For file properties to be saved, enable *Linux extended attributes* in **NSS** by adding the following lines to the `/etc/opt/novell/nss/nssstart.cfg` file:

```
/ListXattrNWMetadata
/CtimeIsMetadataModTime
```

Note that **NSS** supports *Linux extended attributes* starting from **Open Enterprise Server 2.**

For details on *Linux extended attributes*, refer to the **Open Enterprise Server** documentation.

Using drweb-nss-qcontrol

To manage quarantined files and search in the directory, use **`drweb-nss-qcontrol`** utility. On its startup, the utility connects to **Dr.Web Agent** and receives its configuration if the `--agent` command line parameter is not empty.

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

- `-h [ --help ]` – outputs information about supported command line parameters
- `-v [ --version ]` – outputs version number
- `-l [ --level ] <level>` – verbosity level for logging information (logging settings are specified in the `[Logging]` configuration file section, as for **NSS Daemon**)
- `-i [ --ipc-level ] <level>` – verbosity level for logging IPC library information
- `--log-filename <filename>` – name of the log file
- `--agent <address>` – **Dr.Web Agent** address used by other components to receive configuration. If not specified, a component does not request its configuration from **Dr.Web Agent** and operates with the command-line parameters and defaults
- `--timeout <time>` – maximum time to wait for reply from **Dr.Web Daemon** and configuration from **Dr.Web Agent**.
- `--show <regexp>` – outputs general information on files in **Quarantine**. `<regexp>` specifies a regular expression for names of required files. Information is displayed in the following format:

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

  where:

  - `NAME` – name of the file in **Quarantine**
  - `PATH` – full path to the original file location
  - `SIZE` – file size in bytes
  - `TIME` – local time when the file was quarantined
  - `VIRUSES` – comma-separated list of all viruses detected in the file
  - `CODE` – **Dr.Web Daemon** return code in hexadecimal form
  - `ATTRIBUTES` – original file attributes in octal form (the attributes are reset when restoring the file).

  Saved **NSS** attributes are not output.

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 files matching the specified regular expression from the **Quarantine** directory.

  Example:

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

  As a result, all files will be removed from the **Quarantine** directory.

- `--restore <regexp>` – attempts to restore files matching the specified regular expression to their original location (or to another directory if `--restore-dir` command line parameter is specified). All file attributes are restored. File attributes supported only by **NSS** are restored if the target directory is located in the **NSS** volume.

  If a file to be restored is infected, specify the path to its original location in the **ExcludedPaths** parameter in the `[NSS]` section and ensure that it is not specified in **IncludedPaths** parameter. Otherwise, **NSS Daemon** immediately detects the infected file and returns it to the **Quarantine** directory. If after a virus database update a quarantined file is considered not infected, you can restore the file to its original location by specifying the `--rescan` command line parameter. At that, if in the original directory another file with the same name is located, the user is asked whether or not to replace it with the restored file.

  Example:

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

  The utility attempts to restore all files `eicar` in their names to the original location.

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

  Example:

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

  The utility attempts to restore all files containing '`eicar`' in their names to the specified directory "`sample/directory`". If this directory is not in the **NSS** volume, file attributes supported only by **NSS** are not restored.

- `--answer <answer>` – specifies automatic reply whether or not to replace the file when the `--restore` action is applied.

  Example:

  ```
  drweb-nss-qcontrol --restore eicar.comf8JRCG --answer yes
  ```

  The utility attempts to restore all files with names containing `eicar` to their original location overwriting existing files automatically.

- `--rescan <regexp>` – sends all files with names matching regular expression to **Dr.Web Daemon** for rescanning. If after rescanning the file is considered not malicious, it will be automatically restored.

  You can use this parameter to enable automatic restore of "cleaned" files from the **Quarantine** directory. Add the similar line to **crontab** (rescan quarantined files every 30 minutes and restore "clean" files. If another file with the same name already resides in the original directory, it is not overwritten):

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

Configuration File

NSS Daemon settings are specified in the `/etc/drweb/drweb-nss.conf` configuration file.

1. Description of the configuration file structure and parameter types is provided in the Configuration Files section.
2. The `drweb-nss.conf` configuration file contains the following sections:

- [General] – general settings of **NSS Daemon** operation
- [Logging] – logging settings
- [NSS] – settings that manage file scanning and interaction with NSS file system
- [DaemonCommunication] – settings that manage interaction with **Dr.Web Daemon**
- [Actions] – actions applied upon detection of a threat
- [Stat] – settings that manage gathering and sending statistics of anti-virus protection
- [Quarantine] – **Quarantine** settings
- [Notifications] – notification settings.

[General] Section

In the [`General`] section, **NSS Daemon** general settings are specified.

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 cannot be changed by `SUGHUP` signal; for that purpose, module restart is required. Default value: **`BaseDir`** `= /var/drweb`
`MaxTimeoutForThreadActivity = {time}`	Maximum time for a thread to close. This parameter is used on system restart or shutdown. Total time for the system to shut down is calculated as follows: number of pools and the **`MaxTimeoutForThreadActivity`** parameter value are multiplied together, and then a certain 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

In the [`Logging`] section, logging settings are specified. Logging is performed for all main modules of **Dr.Web for Novell Storage Services**.

Parameter	Description
	`[Logging]` section

Parameter	Description
`Level = {log level}`	Log verbosity level. You can specify one of the following levels: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`Level`** `= Info`
`IpcLevel = {log level}`	Log verbosity level for IPC library. You can specify one of the following levels: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`IpcLevel`** `= Alert`
`SyslogFacility = {syslog label}`	Facility label for logging with the `syslog` service. Default value: **`SyslogFacility`** `= Daemon`
`FileName = {syslog | path to file}`	Path to the log file. You can specify `syslog` as a log file name to enable logging by `syslogd` system service. In this case, you must also specify the **`SyslogFacility`** parameter value. Default value: **`FileName`** `= syslog`

[NSS] Section

In the `[NSS]` section, settings for integration with the **NSS** file system are specified.

Parameter	Description
	`[NSS]`
`NSSVolumesMountDir = {path to directory}`	Path to the directory where all NSS volumes are mounted. Specify NSS volumes that must be protected from viruses in the `ProtectedVolumes` parameter value. Default value: **`NSSVolumesMountDir`** `= /media/nss`
`ProtectedVolumes = {list of volumes}`	List of NSS volumes to be protected from viruses. The items in the list must be separated by commas. If the parameter value is empty, all subdirectories specified in the **`NSSVolumesMountDir`** parameter are protected. If some of the listed directories are not NSS volumes, **NSS Daemon** fails to initialize.

Parameter	Description
	Default value: **ProtectedVolumes** =
CheckPoolOptions = {Pool Settings}	Settings of a thread pool that processes scanning tasks. Default value: **CheckPoolOptions** = {2-20}
HeuristicAnalysis = {logical}	Enables or disables heuristic analysis. The detection method used by the *heuristics analyzer* is based on certain knowledge about the attributes that characterize malicious code. Each attribute or characteristic has a weight coefficient that determines the level of its severity and reliability. Depending on the sum weight of a file, the *heuristics analyzer* calculates the probability of unknown virus infection. As with any system of hypothesis testing under uncertainty, the *heuristics analyzer* may commit type I or type II errors (i.e., it may omit viruses or raise false alarms). Note that object detected by the *heuristic analyzer* are treated as suspicious. Default value: **HeuristicAnalysis** = Yes
MaxFileSizeToScan = {Size}	Maximum size of a file that can be scanned by **Dr.Web Daemon**. If the file size is greater than this value, the file is not scanned. If the parameter value is set to 0, file size is not limited. Default value: **MaxFileSizeToScan** = 0b
IncludedPaths = {list of paths}	List of relative paths that are scanned for viruses, regardless of the **ExcludedPaths** parameter value. All paths must be specified relatively to the directory set in the **NSSVolumesMountDir** parameter value: at first, the volume is indicated an then its subdirectories and files. The specified paths must be absolute (that is, without symbols that substitute for the current or parent directory: "." and ".."). Default value: **IncludedPaths** =
ExcludedPaths = {list of paths}	List of relative paths to files that are not scanned unless they are specified in the **IncludedPaths** parameter. All paths must be specified relatively to the directory set in the **NSSVolumesMountDir** parameter: at first, the volume is indicated an then its subdirectories and files. The specified paths must be absolute (that is, without symbols that substitute for the current or parent directory: "." and ".."). Default value: **ExcludedPaths** =

[DaemonCommunication] Section

In the `[DaemonCommunication]` section, settings that configure interaction between **NSS Daemon** and **Dr.Web Daemon** are specified.

Parameter	Description
`[DaemonCommunication]`	
`Address = {weighted addresses list}`	Sockets used by **NSS Daemon** for interaction with **Dr.Web Daemon**. At least one valid address must be specified. Addresses are specified in the following format: `ADDRESS WEIGHT`, where `ADDRESS` is a socket address specified in the standard format (UNIX or TCP socket), and `WEIGHT` is an optional numeric value between 0 and 100. Weight determines a relative work load on a certain host in the network. The greater the value is specified, the greater the load on the server is. If addresses have the same weight, they are considered equal and receive the same number of requests. If 0 is set for an address, it is considered a backup address and it receives requests only if transmission to other addresses with weights greater than 0 failed. When assigning a weight value, consider resources available on the corresponding server. **Examples**: In the following example, only the path to a PID file is specified: **`Address = pid:/var/drweb/run/drwebd.pid`** In the following example, multiple addresses and their weights are specified: **`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 time to wait for **Dr.Web Daemon** to execute a command. If the value is set to 0, the wait time is not limited. Default value: **`Timeout`** `= 2m`

[Actions] Section

In the `[Actions]` section, you can specify actions applied upon detection of a threat or occurrence of an error. The following actions are available:

- `pass` – pass the file;
- `cure` – attempt to cure an infected file. If the file cannot be cured, an action specified in the **`Incurable`** parameter is applied;
- `report` – only send notification (see description of the `[Notifications]` section);
- `quarantine` – move the file to the **Quarantine** directory;
- `remove` – remove the file.

Information on every applied action is logged. If enabled in the [Notifications] section, notification on every applied action, except `pass`, is sent.

Parameter	Description
	`[Actions]`
`Infected = {action}`	Reaction to an object infected with a known virus. You can specify one of the following actions: `remove, quarantine, cure` Default value: **`Infected`** `= cure`
`Suspicious = {action}`	Reaction to a suspicious object that can be infected with an unknown virus (according to heuristics analysis results). You can specified one of the following actions: `remove, quarantine, pass, report` Default value: **`Suspicious`** `= quarantine`
`Incurable = {action}`	Reaction to to an infected object which cannot be cured (only if **`Infected`** `= Cure`). You can specify one of the following actions: `remove, quarantine` Default value: **`Incurable`** `= quarantine`
`Adware = {action}`	Reaction to an object containing an advertising program (adware). You can specify one of the following actions: `remove, quarantine, pass, report` Default value: **`Adware`** `= quarantine`
`Dialers = {action}`	Reaction to an object containing a dialer program. You can specify one of the following actions: `remove, quarantine, pass, report` Default value: **`Dialers`** `= quarantine`
`Jokes = {action}`	Reaction to an object containing a joke program. You can specify one of the following actions: `remove, quarantine, pass, report` Default value: **`Jokes`** `= report`
`Riskware = {action}`	Reaction to riskware (programs that can be used to harm the system). You can specify one of the following actions: `remove, quarantine, pass, report` Default value: **`Riskware`** `= report`
`ArchiveRestriction = {action}`	Reaction to an archive that cannot be scanned by **Dr.Web Daemon** because a threshold value specified in the main configuration file **`drweb32.ini`** was exceeded. You can specify one of the following actions: `remove, quarantine, pass, report`

Parameter	Description
	Default value: **ArchiveRestriction** = quarantine
Hacktools = {action}	Reaction to a program used for hacking. You can specify one of the following actions: remove, quarantine, pass, report Default value: **Hacktools** = report
SkipObject = {action}	Reaction to an object that cannot be scanned by **Dr.Web Daemon** You can specify one of the following actions: remove, quarantine, pass, report Default value: **SkipObject** = report
DaemonError = {action}	Reaction to an object that caused errors during scanning. You can specify one of the following actions: remove, quarantine, pass, report Default value: **DaemonError** = quarantine
LicenseError = {action}	Reaction to an object during scanning of which a license error occurred. You can specify one of the following actions: remove, quarantine, pass, report Default value: **LicenseError** = report
ProcessingError = {action}	Reaction to an object during processing of which an error in **NSS Daemon** operation occurred. You can specify one of the following actions: remove, quarantine, pass, report Default value: **ProcessingError** = report

[Stat] Section

In the [Stat] section, you can specify settings for statistics gathering.

Parameter	Description
	[Stat]
SendToAgent = {logical}	Enables or disables sending statistics on **NSS Daemon** operation to **Dr.Web Agent**. If the parameter value is set to No, statistics is not gathered. Default value: **SendToAgent** = yes
SendPeriod = {time}	Time interval to send statistics to **Dr.Web Agent**. Default value: **SendPeriod** = 5m

[Quarantine] Section

In the `[Quarantine]` section, you can specify **Quarantine** settings.

Parameter	Description
	`[Quarantine]`
`Path = {path to directory}`	Path to the **Quarantine** directory. **NSS Daemon** module must have permissions to create, change, delete and read files in this directory. Default value: **`Path`** = `/var/drweb/infected/nss`
`FilesMode = {numerical value}`	Permissions set for files that are moved to **Quarantine**. Default value: **`FilesMode`** = `0660`

[Notifications] Section

In the `[Notifications]` 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 external program execution after an action (`remove`, `quarantine`, `cure`, `report`) is applied to a file. After the command is executed, information on the event is logged. A thread executing this command waits for it to terminate, and if the return code is not zero, the corresponding message is logged. You can use the following macros in this command: • **`$HOSTMASTER$`** - **`Hostmaster`** parameter value • **`$REASON$`** - name of the event that cause the command execution • **`$ACTION$`** - name of the applied action • **`$VERSION$`** - current product version • **`$FILE$`** - full path to the file which caused the event • **`$SIZE$`** - size (in bytes) of the file which caused the event • **`$TIME$`** - local server time when the command was executed • **`$DAEMON_REPORT$`** - **Dr.Web Daemon** report received after the file was processed. The report can be empty. Lines in report are delimited with a line feed character • **`$VIRUSES$`** - list of viruses detected during scanning. The list can be empty. Items in the list are separated by commas. **Example**: (must be specified on a single 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 this exapmle, the KDE environment, upon every event (for example, scan errors, malware detection) a pop-up notification appears.

Parameter	Description
	Default value: **ExternalProgram** =
SendMail = {logical}	Enables or disables sending of e-mail messages after `remove`, `quarantine`, `cure` or `report` applied to a file. Command to sending an e-mail notification is executed after an action is applied but before it is logged. An e-mail notification is sent to the address specified in the **Hostmaster** parameter value. Templates for the notifications are taken from the directory specified in **Templates** parameter. Default value: **SendMail** = No
Templates = {path to directory}	Path to the directory containing notification templates. Currently, only **email.templ** template must be located in this directory. For that template, you can use macros listed in the **ExternalProgram** parameter value. Default value: **Templates** = /etc/drweb/templates/nss
Hostmaster = {e-mail address}	E-mail address where e-mail notifications are sent. Default value: **Hostmaster** = root@localhost
MailCommand = {String}	Shell command executed to send a notification to the administrator. Default value: **MailCommand** = "/usr/sbin/sendmail -i -bm -f drweb-nss -- %s"

Dr.Web Updater

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

Dr.Web Updater requires installed **Perl** 5.8.0 or later.

For **Fedora** OS 19.0 and 20.0, **Dr.Web Updater** requires additionally the following **Perl** libraries: **perl-Data-Dumper** and **perl-Sys-Syslog** (use version of the libraries with suffix `.686` – for Intel x86 platform; and with suffix `.x86_64` – for amd64 platform).

Dr.Web Updater settings are located in the `[Updater]` section of the `drweb32.ini` configuration file in `%etc_dir` directory. To use an alternative configuration file, specify the full path to it with a command line parameter on the startup.

To run the script, use the following command:

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

For details on allowed parameters, see Command Line Parameters.

In the standard mode, updates are downloaded and installed automatically under the `drweb` user.
Do not start updating under the `root` superuser as this results in changing the ownership of updated files to `root` superuser and may cause an error on attempt to update them automatically in the future.

Updating Anti-Virus and Virus Databases

To provide reliable protection, **Dr.Web for Novell Storage Services** requires regular updates to virus databases.

Dr.Web for Novell Storage Services virus databases are stored as files with the *.`vdb` extension. Update servers of **Dr.Web Global Updating System** (**Dr.Web GUS**) can also store them within 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 a version number of an anti-virus engine, and YY is a sequential number, starting with 00 (for example, the first regular update for version 6.0 is named `drw60000.vdb`).

"Hot" updates are issued daily or even several times a day to provide effective protection against new viruses. These updates are installed over the old ones: that is, a previous `drwtoday.vdb` file is overwritten. When a new regular update is released, all records from `drwtoday.vdb` are copied to `drwXXXYY.vdb`, and a 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 an update to the main virus databases, place the corresponding file to the directory with **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 and others) are supplied in two additional files - `drwrisky.vdb` and `drwnasty.vdb` - with the structure similar to virus databases. These files are also regularly updated: `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 anti-virus techniques are developed), new versions of the anti-virus package are released, containing the updated algorithms, implemented in the anti-virus 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 after an upgrade of a package version, new databases can be linked 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 **Dr.Web Engine.**

Being regularly updated, 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 with **Dr.Web Updater** module (`%bin_dir/update.pl`). After installation, a user crontab file (`/etc/cron.d/drweb-update`) is automatically created to run **Updater** every 30 minutes. That ensures regular updates and maximum protection. You can modify this file to change update period.

Cron Configuration

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

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

Please note that by default the **cron** daemon launches **Dr.Web Updater** once in 30 minutes (at the 0 and 30 minutes of every hour). This may result in increased load on the **Dr.Web GUS** update servers and cause update delays. To avoid such situation, it is recomended to change default values to arbitrary.

Command Line Parameters

- `--help` – shows brief help.
- `--ini` – specifies another (not default) configuration file to be used. To use another configuration file, specify the full path to it with the `--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` – temporarily overrides value of the **`Section`** parameter on **Updater** startup. The new specified value is used until next start of the script. Possible values: `scanner` or `daemon`.

  Example:

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

- `--components` – displays a list of all product components available for update.

  Example:

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

- You can also use the command line parameter `--not-need-reload`:
  - if this parameter is not specified, all daemons (**Dr.Web Daemon** for **Dr.Web for Novell Storage Services**) which components were updated, removed, or added are restarted after **`update.pl`** script finishes;
  - if the `--not-need-reload` parameter is specified without any value, after the **`update.pl`** script finishes no daemon of **Dr.Web for Novell Storage Services** is restarted;
  - if some daemon names are specified as the `not-need-restart` value, the corresponding daemons are not restarted after the **`update.pl`** script finishes. Names of non-restarted daemons must be separated be commas and listed without white spaces. The names are 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 to selected components of your **Dr.Web for Novell Storage Services**.

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

Example:

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

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

If updates to a component are blocked, that component is marked as *frozen*. Frozen components are not updated when **Dr.Web Updater** is started.

Blocking updates

To block updates for specific component, use the `--freeze=<components>` command-line parameter, where `<components>` is a comma separated list 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 enable updates for a frozen component, use the `--unfreeze=<components>` command-line parameter, where `<components>` is a comma separated list 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 **Dr.Web for Novell Storage Services** components are being updated, **Dr.Web Updater** saves their back-up copies to the working directory. It enables you to restore any component to its previous state if any problem occurs during an update.

To restore component to its previous state, use the `--restore=<components>` command line parameter, where `<components>` is a comma separated 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
```

Restored components are automatically frozen. To enable updates for a restored component, unfreeze it.

Configuration

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

Section [Updater]

UpdatePluginsOnly = {logical}	If `Yes` value is specified, **Dr.Web Updater** does not update **Dr.Web Daemon** and **Dr.Web Scanner**. It updates only the plug-ins. Default value: **UpdatePluginsOnly** = No

Section = {Daemon | Scanner}	Specifies the section of configuration file where **Dr.Web Updater** takes the settings, such as a path to the key file, paths to virus databases and others. Possible values: `Scanner, Daemon`. Value of this parameter can be temporarily overriden by the `--what` command line parameter. The specified value is used until the next start of the script. Default value: **Section** = Daemon
ProgramPath = {path to file}	Path to the executable file of **Dr.Web Daemon** or **Dr.Web Scanner**. It is used by **Dr.Web Updater** to get the product version. Default value: **ProgramPath** = %bin_dir/drwebd
SignedReader = {path to file}	Path to the program which is used to read digitally signed files. Default value: **SignedReader** = %bin_dir/read_signed
LzmaDecoderPath = {path to directory}	Path to the directory that contains a program used for unpacking of lzma-archives. Default value: **LzmaDecoderPath** = %bin_dir/
LockFile = {path to file}	Path to the 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** outputs an 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`) with the list of **Dr.Web GUS** servers. **Dr.Web Updater** selects a server from this list in random order to download updates. For details on downloading updates, see Updating Process. This file is signed by **Doctor Web** and must not be modified by a user. The file 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 **Dr.Web GUS** servers. **Dr.Web Updater** also selects a server from this list in random order to download updates. For details on downloading updates, see Updating Process. This file is signed by **Doctor Web** and must not be modified by a

	user. It is updated automatically. Default value: **CustomDrlFile** = %var_dir/bases/custom.drl
FallbackToDrl = {logical}	Allows using the file specified by **DrlFile** when connection to one of the servers listed in **CustomDrlFile** failed. If the parameter value is No, the file specified in **DrlFile** is not used. If the file specified in **CustomDrlFile** does not exist, the file specified in **DrlFile** is used regardless of the **FallbackToDrl** parameter value. For details on downloading updates, see Updating Process. Default value: **FallbackToDrl** = Yes
DrlDir = {path to directory}	Path to the directory that contains drl files with lists of **Dr.Web GUS** servers for each plug-in. These files are signed by **Doctor Web** and must not be modified by a user. Default value: **DrlDir** = %var_dir/drl/
Timeout = {numerical value}	Maximum wait time for downloading updates from the selected **Dr.Web GUS** server, in seconds. Default value: **Timeout** = 90
Tries = {numerical value}	Number of attempts by **Dr.Web Updater** to establish 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 to access the used proxy server (if it requires authentication). Default value: **ProxyLogin** =
ProxyPassword = {string}	The password to access the used proxy server (if it requires authentication). Default value: **ProxyPassword** =
LogFileName = {syslog | file name}	Path to the log file name. You can specify syslog as a log file name and logging will be

	performed by `syslogd` system service. 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. The following levels are allowed: • `Quiet` • `Error` • `Warning` • `Info` • `Debug` • `Verbose` Default value: **`LogLevel`** = `Info`
`BlacklistPath` = `{path to directory}`	Path to the 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 left before license expiration during which **Dr.Web Updater** is attempting to update license key file. Default value: **`ExpiredTimeLimit`** = `14`
`ESLockfile` = `{path to file}`	Path to the lock file. If the lock file exists, **Dr.Web Updater** can not be automatically initialized by `cron` daemon. Default value: **`ESLockfile`** = `%var_dir/run/es_updater.lock`

Updating Procedure

Updating is performed in the following stages:

1. **Dr.Web Updater** reads the configuration file (`drweb32.ini` by default, or specified with the `--ini` command line argument).
2. **Dr.Web Updater** uses parameters from the `[Updater]` section of the configuration file (see the description above) as well as the following parameters: **`EnginePath`**, **`VirusBase`**, **`UpdatePath`** and **`PidFile`**.

3. **Dr.Web Updater** selects **Dr.Web GUS** server for downloading updates. The server is selected in the following way:
   - Reading of the files which contain lists of update servers. The filenames are specified in the **DrlFile** and **CustomDrlFile** parameters;
   - If both files are not accessible, updating process stops and terminates;
   - If only one of the files is accessible, it is used regardless of the value specified for the **FallbackToDrl** parameter;
   - If both files are accessible, **Dr.Web Updater** uses the file specified in the **CustomDrlFile** parameter;
   - If it is impossible to connect to any of the servers from this file (specified in **CustomDrlFile**), and the **FallbackToDrl** value is set to `Yes`, **Dr.Web Updater** tries to establish connection with the servers from the file specified in the **DrlFile** parameter. If the connection fails, the updating process stops and terminates.
4. **Dr.Web Updater** tries to connect to servers from the selected file in random order until connection is established (**Dr.Web Updater** waits for the server to respond during the period specified in the **Timeout** parameter).
5. **Dr.Web Updater** requests the list of available updates from the selected **Dr.Web GUS** server and then requests the corresponding lzma archives. If the archives are not available on the server, the updates are downloaded as `vdb` files. To unpack lzma-archives, **lzma** utility is used. Path to the directory with the utility is specified in the **LzmaDecoderPath** parameter.
6. After updates are unpacked, they are saved to the corresponding directories as described in Updating.

Dr.Web Monitor

Dr.Web Monitor is 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 suite components as well as restart of any component if it is operating abnormally. **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 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 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 a corporate or private **Anti-virus network** managed by **Dr.Web Enterprise Security Suite**. To operate in the central protection mode, it is not required 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 following modes:

- **Standalone mode** when a protected computer is not included in an anti-virus network and is managed locally. In this mode, configuration files and key files reside on local drives, **Dr.Web Monitor** is fully controlled from the protected computer, and all modules start in accordance with the settings specified in the **Dr.Web Monitor** configuration file.
- **Enterprise mode** (or **central protection mode**) when protection of the local computer is managed from the central protection server. In this mode, some features and settings of **Dr.Web for Novell Storage Services** can be modified and blocked for compliance with a general security policy (for example, corporate 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 enable 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 value to `Yes`.

   In the central protection mode, some features and settings of **Dr.Web for Novell Storage Services** can be modified or blocked for compliance with the general 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.

For **Dr.Web for Novell Storage Services** to fully support the central protection mode, also enable **Dr.Web Agent** to operate in the Enterprise mode. For details, see Operation Mode of **Dr.Web Agent**.

To enable standalone mode

1. Ensure that all modules that you want **Dr.Web Monitor** to start are listed in the **RunAppList** parameter in the [Monitor] section of **Dr.Web Monitor** configuration file (by default, %etc_dir/monitor.conf). The modules must be installed and configured properly.
2. In the [Monitor] section of **Dr.Web Monitor** configuration file, set the **UseEnterpriseMode** parameter value 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 access all settings of **Dr.Web for Novell Storage Services** again and configure them.

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

Command Line Parameters

To run **Dr.Web Monitor**, use this 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 the module		
-v	--version	
Description: Show **Dr.Web Monitor** version on the screen and terminate the module		
-u	--update	
Description: Start updating all **Dr.Web for Novell Storage Services** components		
-C	--check-only	
Description: Check correctness of **Dr.Web Monitor** configuration. This parameter cannot be used if a **Dr.Web Monitor** process is already running in the system.		
-A	--check-all	<path to file>
Description: Check correctness of configuration of all **Dr.Web for Novell Storage Services** components		
-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. Use the application name specified in the header of the Application "<application name>" section in the corresponding mmc file (for details, see Interaction with other Suite Components). This parameter cannot be used if a **Dr.Web Monitor** process is already running in the system.		

Example usage:

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

Configuration File

Adjustment of **Dr.Web Monitor** settings is performed in its configuration file `%etc_dir/monitor.conf`.

For general organization concept of **Dr.Web for Novell Storage Services** configuration files, see Configuration Files.

[Logging] Section

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

`[Logging]`

Level = {log level}	**Dr.Web Monitor** log verbosity level. The following levels are available: • Quiet • Error • Alert • Info • Debug Default value: **Level** = Info
IPCLevel = {log level}	Log verbosity level for IPC library. The following levels are available: • 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 the log file. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. In this case, you must also specify the **SyslogFacility** parameter. Default value: **FileName** = syslog

[Monitor] Section

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

`[Monitor]`

RunForeground = {logical}	**Yes** value forbids **Dr.Web Monitor** to operate in `daemon` mode. This option can be used by some monitoring utilities (for example, **daemontools**). Default value: **RunForeground** = No
User = {text value}	Name of the user whose privileges are used by **Dr.Web Monitor.** 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 the directory of a file where information on **Dr.Web Monitor** process identifier (PID) is written upon the module startup. Default value: **PidFileDir** = %var_dir/run/
ChDir = {path to directory}	Change of working directory upon **Dr.Web Monitor** startup. If this parameter is set, **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 the directory where metaconfiguration files reside. These files contain settings defining **Dr.Web Monitor** interaction with other **Dr.Web** components. Metaconfiguration files are provided by **Dr.Web** developers and do not require editing. Default value: **MetaConfigDir** = %etc_dir/monitor/
Address = {address}	Socket used by **Dr.Web Monitor** to receive control signals from other **Dr.Web** components. Default value: **Address** = local:%var_dir/ipc/.monitor
Timeout = {numerical value}	Maximum time (in seconds) to establish connection between **Dr.Web Monitor** and other **Dr.Web** components. Default value: **Timeout** = 5
TmpFileFmt = {text value}	Name templates for **Dr.Web Monitor** temporary files. Template format: `path_to_file.XXXXXX` where `X` is a 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**; use comma as a delimiter. Please note that this parameter is not modified upon uninstalling a **Dr.Web** component. You must manually remove the uninstalled component from this parameter value. Otherwise, **Dr.Web Monitor** will not be able to run and start other **Dr.Web** components. Default value: **RunAppList** = AGENT
UseEnterpriseMode = {logical}	If the value is set to Yes, **Dr.Web Monitor** receives the list of modules to be started from **Dr.Web Agent** rather than from the **RunAppList** parameter value. Default value: **UseEnterpriseMode** = No
RecoveryTimeList = {numerical values}	Time intervals between attempts to restart components that are not responding (in seconds). This parameter can have multiple values, separated by commas. First attempt to restart a component is made after a period of time specified in the first parameter value, second attempt – using the 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 other addresses (not only to root@localhost), you need to specify the addresses 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 the **Address** parameter value from **Dr.Web Agent** configuration file). Default value: **AgentAddress** = local:%var_dir/ipc/.agent
AgentResponseTime = {numerical value}	Maximum time to wait a response from drweb-agent module in seconds. If **Dr.Web Agent** does not respond during this time period, **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 started with the default settings, the following actions are performed:

1. **Dr.Web Monitor** searches for and loads its configuration file. If the configuration file is not found, loading process stops;
2. **Dr.Web Monitor** starts operating in the `daemon` mode. So, information about loading problems cannot be output to the console and, thus, is logged to the file;
3. Socket for **Dr.Web Monitor** interaction with other **Dr.Web for Novell Storage Services** modules is created. If a TCP socket is used, several connections can be established (loading process continues if at least one connection is established). If a 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 a socket cannot be created, loading process stops;
4. PID-file with information on **`drweb-monitor`** process identifier is created. If the PID-file cannot be created, loading process stops;
5. **`drweb-monitor`** module starts other suite components. If a module cannot load, **Dr.Web Monitor** tries to restart it. If all **Dr.Web Monitor** attempts to start the module failed, **Dr.Web Monitor** unloads all previously loaded modules and terminates. **Dr.Web Monitor** reports problems connected with the modules startup in one of the available ways (logging to the file, notifying via email, startup of a custom program). Notification methods used for various modules are set in the **Dr.Web Monitor** meta-configuration file (.mmc).

To start **Dr.Web Monitor** in the automatic mode, do one of the following:

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

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

Location of the enable files depends on **Dr.Web for Novell Storage Services** installation type:

- Installation from the **universal package for UNIX systems**:
  Files will be saved to `%etc_dir` directory and have the following names
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation from **native DEB packages**:
  Files will be saved to `/etc/defaults` directory and have the following names
  `drwebd`,
  `drweb-monitor`.
- Installation from **native RPM packages**:
  Files will be saved to `/etc/sysconfig` directory and have the following names
  `drwebd.enable`,
  `drweb-monitor.enable`.

Interaction with Other Suite Components

Interaction with other suite components is performed with the use of **Dr.Web Monitor** meta-configuration files (`mmc` files). These files are included in packages of those products which can interact with **Dr.Web Monitor** and reside in reside in the directory specified in the **`MetaConfDir`** parameter (by default - `%etc_dir/monitor`). The files contain information on component composition, location of binary files, their launch order and startup options. Usually, one file contains information on one component and name of the file matches to the name of the **Dr.Web for Novell Storage Services** component.

Each component is described in the `Application` section with the corresponding name. At the end of

the section, `EndApplication` must be specified.

The following parameters must be present in the component description:

- **`FullName`** – full name of the component.
- **`Path`** – path to the binary files.
- **`Depends`** – names of the components which must be started before the described component. For example, `AGENT` component must be started before **Dr.Web Daemon,** therefore in the `mmc` file for **Dr.Web Daemon** **`Depends`** parameter has the `AGENT` value. If there are no dependencies, this parameter can be skipped.
- **`Components`** – list of binary files of modules started together with the component. Modules are started in the same order as they are specified in this parameter. For each module the following information must be specified (space separated): command line parameters (can be enclosed in quotation marks), timeouts for startup and stop (`StartTimeout` and `StopTimeout`), notification type and startup privileges. *Notification type* – defines where notifications on component failure are sent. When `MAIL` value is specified, notifications are sent by mail, when `LOG` value is specified, information is only logged to the file. *Startup privileges* – defines a group and a user, whose privileges are used by the component.

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 root:drweb
 EndComponents
EndApplication
```

Example of mmc file for Dr.Web NSS:

```
Application "NSS"
 FullName    "Dr.Web (R) NSS"
 Path        "/opt/drweb/"
 Depends     "AGENT"
 Components
  # name  args  MaxStartTime  MaxStopTime  NotifyType  User:Group
  drweb-nss  local:/var/drweb/ipc/.agent  30  30  MAIL  root:drweb
 EndComponents
EndApplication
```

Dr.Web Agent

Dr.Web Agent is a resident module used to manage settings of **Dr.Web for Novell Storage Services** modules, define anti-virus policy depending on available licenses and collect virus statistics. Statistics, depending on **Dr.Web Agent** operational mode, is sent with the predetermined frequency either to the public server of the company or to the central protection server that works under **Dr.Web Agent**. When **Dr.Web for Novell Storage Services** modules are started or settings are changed, **Dr.Web Agent** sends all necessary configuration to these modules.

Note that **drweb-agent** can operate in enterprise mode only with **Dr.Web ESS** 6. If you want to ensure connection to the central protection server **Dr.Web ESS** 10, install and configure the new agent version, implemented as **drweb-agent10** module. For details on how to install and configure **drweb-agent10**, refer to the Migration to Dr.Web ESS 10 section.

Dr.Web Agent can interact with other modules through exchanging control signals.

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

Please note that when several parameters with the same name are specified in the configuration file, **Dr.Web Agent** unites them in one comma delimited string. You can also use a backslash symbol "\" to define parameter value in several lines. New line after backslash is added to the previous line when **Dr.Web Agent** is reading configuration. Note that using of a space character after a slash is not allowed.

Operation Mode

If necessary, **Doctor Web** can be connected to a corporate or private anti-virus network managed by **Dr.Web Enterprise Security Suite** (**Dr.Web ESS**). To operate in the 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 **Dr.Web Agent** is fully controlled from the protected computer.
- **Enterprise mode** (or central protection mode), when protection of the computer is managed from the 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 (for example, 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.

Note that **drweb-agent** can operate in enterprise mode only with **Dr.Web ESS** 6. If you want to ensure connection to the central protection server **Dr.Web ESS** 10, install and configure the new agent version, implemented as **drweb-agent10** module. For details on how to install and configure **drweb-agent10**, refer to the Migration to Dr.Web ESS 10 section.

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`), adjust the following parameters in the `[EnterpriseMode]` section:

- Set the **PublicKeyFile** parameter value 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 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 value to the IP-address or host name of the **Enterprise Server**.
- Set the **ServerPort** parameter value to the **Enterprise Server** port number.

3. To connect to the central protection server, set the **UserEnterpriseMode** parameter value 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 the central protection server. Your personal key file on the local computer is not used.

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

To enable **Dr.Web for Novell Storage Services** to fully support the central protection mode, set **Dr.Web Monitor** to operate in enterprise mode. For more details, see Operation Mode of **Dr.Web Monitor**.

To use standalone mode

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

   When 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 access all features of **Dr.Web for Novell Storage Services** solutions again and configure them.

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

Using Dr.Web for Novell Storage Services and Dr.Web Anti-virus for Linux together in the central protection mode

Because of the implementation features, **Dr.Web for Novell Storage Services** and **Dr.Web Anti-virus for Linux** cannot be simultaneously operate in the central protection mode if they are both installed on the same computer. To enable **Dr.Web for Novell Storage Services** to operate in the central protection mode, change the operation mode of **Dr.Web Anti-virus for Linux** to the Standalone mode and delete or move to another directory the following files: %etc_dir/agent/drweb-cc.amc and %etc_dir/agent/drweb-spider.amc.

If you want to switch **Dr.Web Anti-virus for Linux** back to the central protection mode later, we recommended to save the files as a back up copy in a directory that is different from %etc_dir/agent. In this case, disable the central protection mode of **Dr.Web for Novell Storage Services**, copy back up copies of drweb-cc.amc and drweb-spider.amc files to the %etc_dir/agent/ directory and follow the instructions provided in the **Dr.Web Anti-virus for Linux** User Manual.

Command Line Parameters

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

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

where the following `parameters` are available:

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and terminate the module		
`-v`	`--version`	
Description: Show **Dr.Web Agent** version on the screen and terminate the module		
`-u`	`--update-all`	
Description: Start updating all **Dr.Web for Novell Storage Services** components		
`-f`	`--update-failed`	
Description: Start updating **Dr.Web for Novell Storage Services** components, updating of which failed in the standard mode		
`-C`	`--check-only`	
Description: Check correctness of **Dr.Web Agent** configuration. This parameter cannot be used if a **Dr.Web Agent** process is already running in the system		
`-c`	`--conf`	`<path to file>`
Description: Enable the module to use the specified configuration file		
`-d`	`--droppwd`	
Description: Discard registration data required to access **Dr.Web Enterprise Server** (username, password). At the next connection attempt, a new process of workstation registration will start.		
`-p`	`--newpwd`	
Description: Change username and password required to access **Dr.Web Enterprise Server**		
`-s`	`--socket`	`<path to file>`
Description: Use the specified socket for interaction with the controlled modules		
`-P`	`--pid-file`	`<path to file>`
Description: Use the specified file as a PID file of **Dr.Web Agent**		
`-e`	`--export-config`	`<application name>`
Description: Export configuration of the specified application to **Dr.Web Enterprise Server**. Use the application name specified in the header of the `Application "<application name>"` section in the corresponding `amc` file (see Interaction with other Suite components). This parameter cannot be used if a **Dr.Web Agent** process is already running in the system or if you want to export **Dr.Web Anti-virus for Linux** configuration.		

Configuration File

Configuration of **Dr.Web Agent** is specified in the following file: `%etc_dir/agent.conf`.

For general organization concept of **Dr.Web for Novell Storage Services** configuration files, see Configuration Files.

[Logging] Section

The `[Logging]` section contains **Dr.Web Agent** logging settings:

`[Logging]`

Level = {log level}	**Dr.Web Agent** log verbosity level. The following levels are available: • Quiet • Error • Alert • Info • Debug Default value: **Level** = Info
IPCLevel = {log level}	Log verbosity level of IPC library. The following levels are available: • Quiet • Error • Alert • Info • Debug Default value: **IPCLevel** = Error
SyslogFacility = {syslog label}	Log type label used by `syslogd` system service. Default value: **SyslogFacility** = Daemon
FileName = {path to file | syslog}	Path to the log file. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. Default value: **FileName** = syslog

[Agent] Section

The `[Agent]` section contains general **Dr.Web Agent** settings:

`[Agent]`

MetaConfigDir = {path to directory}	Name of the directory where meta-configuration files of **drweb-agent** are located.

	These files contain settings of interaction between **Dr.Web Agent** and other modules of the **Dr.Web** suite. Meta-configuration files are provided by **Dr.Web** developers and do not need to be modified. Default value: **MetaConfigDir** = %etc_dir/agent/
UseMonitor = {logical}	Yes value indicates to **drweb-agent** that **Dr.Web Monitor** is used as a part of **Dr.Web for Novell Storage Services**. Default value: **UseMonitor** = Yes
MonitorAddress = {address}	Socket used by **Dr.Web Agent** for interaction with **Dr.Web Monitor** (the parameter value must be the same as the **Address** parameter value in the **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** does not respond during this period, **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}	Name of the file where **Dr.Web Agent** PID is written on **Dr.Web Agent** startup. Default value: **PidFile** = %var_dir/run/drweb-agent.pid

[Server] Section

The [Server] section contains parameters that control interaction of **Dr.Web Agent** with other **Dr.Web for Novell Storage Services** modules:

[Server]

Address = {address}	Socket used by **Dr.Web Agent** to interact with other modules of the suite. You can specify multiple sockets separating them by comma. 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 determines maximum number of simultaneous connections to modules that report virus statistics to **Dr.Web Agent**. The parameter value cannot be changed with 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 the value is set to 0, time for establishing connection is unlimited. Default value: **Timeout** = 15

[EnterpriseMode] Section

The [EnterpriseMode] section contains parameters of **Dr.Web Agent** operation in the **Enterprise** mode:

[EnterpriseMode]

UseEnterpriseMode = {logical}	If the value is set to Yes, **Dr.Web Agent** operates in the Enterprise mode, if the value is set to No - in the Standalone mode. Default value: **UseEnterpriseMode** = No
ComputerName = {text value}	Name of the computer in **Anti-virus network**. Default value: **ComputerName** =
VirusbaseDir = {path to directory}	Path to the directory where virus databases are located. Default value: **VirusbaseDir** = %var_dir/bases
PublicKeyFile = {path to file}	Path to the public key file required to access **Dr.Web Enterprise Server**. Default value: **PublicKeyFile** = %bin_dir/drwcsd.pub
ServerHost = {IP address}	IP address of **Dr.Web Enterprise Server** . Default value: **ServerHost** = 127.0.0.1
ServerPort = {port number}	Number of the port required 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 – force encryption • Possible – encrypt if possible • No – do not encrypt Default value: **CryptTraffic** = possible
CompressTraffic = {Yes | Possible | No}	Compression of traffic between **Dr.Web Enterprise Server** and **Dr.Web Agent**: • Yes – force compression

	• `Possible` – compress if possible • `No` – do not compress Default value: **`CompressTraffic`** = `possible`
`CacheDir` = `{path to directory}`	Path to the 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

The `[StandaloneMode]` section contains parameters of **Dr.Web Agent** operation in the **Standalone** mode:

`[StandaloneMode]`

`StatisticsServer` = `{text value}`	Address (URL) of the virus statistics server If the value is not specified, statistics is not sent. Default value: **`StatisticsServer`** = `stat.drweb.com:80/update`
`StatisticsUpdatePeriod` = `{numerical value}`	Period (in minutes) for statistics updating. Value cannot be less than 5 Default value: **`StatisticsUpdatePeriod`** = `10`
`StatisticsProxy` = `{hostname | IP address}`	IP address or host name of proxy server for sending virus statistics. Please note that if the parameter value is not set, the value of `http_proxy` environment variable is used. **Example**: `StatisticsProxy = localhost:3128` Default value: **`StatisticsProxy`** =
`StatisticsProxyAuth` = `{text value}`	Authentication string (`<username>:<password>`) to access proxy server. **Example**: `StatisticsProxyAuth = test:testpwd` Default value: **`StatisticsProxyAuth`** =
`UUID` = `{text value}`	Unique user ID for the statistics server http://stat.drweb.com/. Please note that this parameter is mandatory for sending statistics. Thus, if you want to enable this option, specify the personal `UUID` as the parameter value (`md5` sum of license key file is usually used as UUID). 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 the list contains more than one path). Default value: **LicenseFile** = %bin_dir/drweb32.key

[Update] Section

The [Update] section contains parameters of **Dr.Web for Novell Storage Services** update 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 (in 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 the root directory. Default value: **RootDir** = /

For more information, see *Administrator Manual* for **Dr.Web ESS**.

Running Dr.Web Agent

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

When **Dr.Web Agent** starts with the 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** is operating within **Anti-virus network**, **Dr.Web Agent** starts in the 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, several connections can be established (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 a socket cannot be created, **Dr.Web Agent** terminates.

Further loading process depends on the selected operation mode.

If **Dr.Web Agent** operates in the **Enterprise mode**:

- **Dr.Web Agent** connects to **Dr.Web Enterprise Server**. If the server is unavailable or authorization process fails during the first connection attempt, **Dr.Web Agent** terminates. If **Dr.Web Agent** worked previously with this server and now the server is temporary unavailable (for example, if any connection problem occurs), **Dr.Web Agent** uses backup copies of configuration files received from the server earlier. These files are encrypted and must not be edited by a user. An attempt to edit the files makes them invalid.
- If the connection is established, **Dr.Web Agent** receives key files and settings from **Dr.Web Enterprise Server**. After all settings and key files are received, **Dr.Web Agent** is fully operational.

If **Dr.Web Agent** operates in the **Standalone mode**, meta-configuration files (.amc) that manage **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 to operate.

Interaction with Other Suite Components

Interaction with other suite components is performed by **Dr.Web Agent** metaconfiguration files (`amc` files). These files contain configuration parameters that are sent to the respective **Dr.Web** modules by **Dr.Web Agent**. The files reside in the directory specified in the **MetaConfDir** parameter (by default - `%etc_dir/agent`). Usually, one file contains configuration parameters of one component and name of the file matches to the name of the **Dr.Web for Novell Storage Services** component.

Each module is described in the `Application` section with the corresponding name. At the end of the section **EndApplication** must be specified.

The following parameters must be present in the module description:

- **id**: identifier of the module in **Dr.Web ESS**.
- **ConfFile**: path to the module configuration file.
- **Components**: description of the modules. At the end of this section, `EndComponents` must be specified. Description of each module must contain the following information: name and list of sections in the configuration file with parameters that are necessary for proper operation. The list of sections and parameters is comma separated.
  To describe individual parameters properly, specify the full path to them (for example, `/Quarantine/DBISettings)`. In the section descriptions, only their names can be specified (for example, `General)`.
  To denote line breaks, a back slash (\) is used.
  If the component requires all settings from the configuration file, you can specify a path "/*" instead of the list of sections and/or parameters.

Example of amc file for NSS Daemon:

```
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** with **Dr.Web Enterprise Security Suite**:

- Setup and initial configuration of **Dr.Web for Novell Storage Services** in the existing **Anti-virus Network** operated by **Dr.Web ESS**;
- Embedding of working UNIX server with already installed and configured **Dr.Web for Novell Storage Services** in the **Anti-virus Network** operated by **Dr.Web ESS**.

To enable **Dr.Web for Novell Storage Services** to work in **Dr.Web ESS** environment, configure **Dr.Web Agent** and **Dr.Web Monitor** components for operation in the `Enterprise` mode, and register the suite on **Dr.Web Enterprise Server**.

According to the connection policy for new working stations (for details, see **Dr.Web Enterprise Security Suite** administrator manual), **Dr.Web for Novell Storage Services** can be connected to **Dr.Web Enterprise Server** in two different ways:

- when a new account is automatically created by the central protection server
- when a new account is created by administrator manually.

Configuring Components to Run in Enterprise Mode

To start the components in the `Enterprise` mode after installation, it is necessary to adjust parameter values in the local configuration files of **Dr.Web Agent** and **Dr.Web Monitor**.

For Dr.Web Agent

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

- **`UseEnterpriseMode`** = `Yes`;
- **`PublicKeyFile`** = `%var_dir/drwcsd.pub` (public encryption key used to access **Dr.Web Enterprise Server**. Administrator must move this file from the corresponding directory of **Dr.Web Enterprise Server** 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 the `[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 a new account is created automatically:

1. On the first run in the `Enterprise` mode, **Dr.Web Agent** 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 the **Approve access manually** mode (used by default; for details, see the administrator manual for **Dr.Web ESS**), system administrator must confirm registration of a new station via **Dr.Web Control Center** web interface in one minute;
3. After the first connection, **Dr.Web Agent** records the hash of the station ID and password into the `pwd` file. This file is created in the directory specified in the **`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** connects to **Dr.Web Enterprise Server**;
5. If you delete the password file, repeated registration request will be sent to **Dr.Web Enterprise Server** on the next **Dr.Web Agent** startup.

Manual Creation of New Account by Administrator

To create a new account manually:

1. Create a new account on **Dr.Web Enterprise Server**: specify the station ID and password (for details, see the administrator manual for **Dr.Web ESS**).
2. Start **Dr.WebAgent** with the **--newpwd** command line parameter (or **-p**) and enter the station ID and password. **Dr.Web Agent** records the hash of station ID and password into the `pwd` file. This file is created in the directory that is specified in the **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** connects to **Dr.Web Enterprise Server**.
4. If you delete the password file, retry registration on the next **Dr.Web Agent** startup.

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

You can configure **Dr.Web for Novell Storage Services** and **Dr.Web Daemon** (anti-virus module included in the standard installation package) via **Dr.Web Control Center**.

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

Export of Existing Configuration to ES Server

You can export configuration from the local computer to **Dr.Web Enterprise Server** automatically when **Dr.Web Agent** is operating in the `Enterprise` mode. To export configuration, use the 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 the System

To start the system:

1. In **Dr.Web Control Center**, open **Dr.Web Monitor** settings and select the **Daemon** and **NSS** check boxes to start the corresponding components;
2. Start **Dr.Web Monitor** on the local computer:

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

Integration with Dr.Web ESS 10

Dr.Web for Novell Storage Services 6.0.2 includes two versions of the **Dr.Web Agent**:

- **Dr.Web Agent**, implemented as **drweb-agent** module, in **enterprise** mode can interact only with **Dr.Web ESS** server version 6.
- **Dr.Web Agent**, implemented as **drweb-agent10** module, in **enterprise** mode can interact only with **Dr.Web ESS** server version 10.

To start using the central protection server **Dr.Web ESS** 10, configure standard integration and also make additional settings.

Configuring connection to Dr.Web ESS 10

As **Dr.Web ESS** does not support management of **Dr.Web Monitor** and **Dr.Web Daemon**, **drweb-agent10** uses two supplementary configuration files in addition to the standard file `%etc_dir/agent.conf`: `es_monitor.conf` and `es_daemon.conf`. They are located in the same directory. These files store configuration for **Dr.Web Monitor** and **Dr.Web Daemon**. The configuration settings will be used for adjusting operation of these modules in **enterprise** mode.

Each file line contains the parameter value of the corresponding module configuration. The format is as follows: `<section>/<parameter> <value>`, where `<section>` is the name of the section from the component configuration file, `<parameter>` is the parameter name, and `<value>` is the value specified for this parameter.

Example (for `es_monitor.conf` file that contains settings for **Dr.Web Monitor** component operation in **enterprise** mode):

```
Monitor/RunAppList DAEMON
```

This line contains the value of **RunAppList** parameter stored in `[Monitor]` section in **Dr.Web Monitor** configuration file. This parameter value is used when the suite is running in **enterprise** mode. In this case, **Dr.Web Monitor** starts only **Dr.Web Daemon**.

Example (for `es_daemon.conf` file that contains settings for **Dr.Web Daemon** component operation in **enterprise** mode):

```
Daemon/MaxCompressionRatio 500
```

This line contains the value of **MaxCompressionRatio** parameter stored in `[Daemon]` section in **Dr.Web Daemon** configuration file. This parameter value is used when the suite is running in **enterprise** mode. In this case, **Dr.Web Daemon** uses 500 as the threshold value of compression ratio.

To connect **Dr.Web for Novell Storage Services** to the central protection server **Dr.Web ESS** 10:

1. Open `agent.mmc` meta-configuration file (used by **Dr.Web Monitor** for communication with **Dr.Web Agent**) and replace the specified binary file name **drweb-agent** with **drweb-agent10**.
2. In `es_monitor.conf` file, specify components to be started int **enterprise** mode. For that purpose, edit the `es_monitor.conf` accordingly. The set of started components must be similar to the set of components started in **standalone** mode (specified as the value of **RunAppList** parameter stored in `[Monitor]` section in **Dr.Web Monitor** configuration file). If more than one component must be started, they are specified as a comma-separated list. Note that white spaces are not allowed. Example:

   ```
   Monitor/RunAppList DAEMON,NSS
   ```

   As the component names, here should be used the names specified in `Application` section of mmc-files.

3. In `es_daemon.conf` file, specify the `root` value for **`Daemon/User`** parameter.
4. If required, configure other parameters in `es_daemon.conf` file that is used by **Dr.Web Daemon** respectively in **enterprise** mode.
5. If **standalone** mode was previously used, switch operation of **Dr.Web Agent** and **Dr.Web Monitor** components to **enterprise** mode by specifying appropriate settings in their configuration files, as described in the Configuring Components to Run in Enterprise Mode section.
6. Restart **Dr.Web Monitor** by using the following command:

```
# service drweb-monitor restart
```

Gathering Virus Statistics

Dr.Web Agent receives statistics on computer threats from the controlled modules and sends it either to the official **Doctor Web** statistics website: http://stat.drweb.com/ (if the Internet connection is available) or to **Dr.Web ESS** (if **Dr.Web Agent** is operating in the Enterprise mode).

Dr.Web Agent needs the *unique user identifier* (UUID) to connect to this website. By default, MD5 hash of the key file is used as a UUID. Also you can get a personal UUID from **Doctor Web Technical Support**. In this case, specify your UUID explicitly in the **Dr.Web Agent** configuration file (`StandaloneMode]` section).

Statistics is gathered only for those **Dr.Web** modules that receive settings from **Dr.Web Agent**. Instructions on how to set up interaction with **Dr.Web Agent** are given in the sections describing the modules.

On the statistics website (at http://stat.drweb.com/), you can view aggregate statistics on computer threats both for a given server and for all servers supported by **Dr.Web Anti-virus for UNIX** or by **Dr.Web for Novell Storage Services** with an anti-virus plug-in. **Dr.Web Agent** can simultaneously process statistics on computer threats from several different **Dr.Web** products which are able to interact with **Dr.Web Agent**.

If **Dr.Web Agent** is operating in the Enterprise mode, you can view statistics on the special page of **Dr.Web Control Center**. In this case, statistics gathered by **Dr.Web Enterprise Server** is also sent to the **Doctor Web** statistics server as a summary of the **Anti-virus network** statistics.

Statistics is available in both HTML and XML formats. The second format is convenient if you plan to publish this statistics on another website, since data in the XML format can be transformed according to the website concept and design.

To view 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.

Appearance of the webpage can differ depending on the used browser.

The following figure shows threats statistics page.

Figure 15. Computer threats statistics

You can change search options and repeat the search. To do this:

1. Select either **Mail** or **Files** check boxes to get statistics on computer threats detected in emails or files.
2. In the drop-down lists for **Start date** and **End date**, select **start/end date** and **time** for the required period.
3. In the **Top** field, enter the required number of rows in the statistics table (most frequently detected threats will be shown).
4. Click **Query**. The file with aggregate statistics in the XML format 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>7.86446592583328</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 shown in the statistics table (number of rows);
- `updatedutc` – last statistics update time;
- `vname` – threat name;
- `place` – place of the virus in the statistics;
- `percents` – percentage of the total number of detections.

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

To get personalized threat statistics

Visit one of the following webpages:

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

The *<UUID>* in both cases stands for the MD5 hash 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 shown in the table (number of rows);
- `user` – user identifier;
- `lastdata` – time when user last sent data to the server;
- `vname` – threat name;
- `place` – threat place in the statistics;
- `caught` – number of detections of the certain threat;
- `percents` – percentage of the total number of detections.

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

Dr.Web Daemon

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

Dr.Web Daemon is always running and has simple and intelligible protocol for sending scanning requests, which makes it a perfect solution to be used as an anti-virus filter for file servers. **Dr.Web for Novell Storage Services** is a ready-made solution for integrating **Dr.Web Daemon** with NSS file system.

Note that **Dr.Web Daemon** cannot scan the contents of the encrypted files because in this case it is necessary to know the password that been used for encryption. So, these files will be passed without the scan, and for the client application the special return code will be returned.

Command-Line Parameters

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

```
drwebd [parameters]
```

where the following `parameters` are available:

Short case	Extended case	Arguments
`-h, -?`	`-help, --help`	
Description: Show information about supported command line parameters on the screen and terminate the module		
`-a`		`<Agent socket address>`
Description: Start **Dr.Web Daemon** in the 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		
	`--foreground`	`<yes|no>`
Description: Operation mode of **Dr.Web Daemon**. If `yes` is specified, **Dr.Web Daemon** is a foreground process. Otherwise (`no`), **Dr.Web Daemon** is a background process		
	`--check-only`	`<command line parameters for checking>`
Description: Check **Dr.Web Daemon** configuration correctness on startup. If any command line parameter is specified, correctness of the value is also checked		
	`--only-key`	
Description: On startup, **Dr.Web Daemon** receives from **Dr.Web Agent** only the license key file		

Running Dr.Web Daemon

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

- Search and load of the configuration file. If the configuration file is not found, loading of **Dr.Web Daemon** terminates. Path to the configuration file can be specified on startup with the `-ini` command line parameter: `{path/to/your/drweb32.ini}`, otherwise, the default value (`%etc_dir/drweb32.ini`) can be used. On startup, correctness of several configuration parameters is checked, and if a parameter value is incorrect, the default parameter value is set;
- Creation of a log file. A user account under which **Dr.Web Daemon** is started must have appropriate privileges to write to the log file directory. Users do not have write permission for the default log directory (`/var/log/`). Therefore, if the **`User`** parameter is specified, adjust the **`LogFileName`** parameter and provide alternative log file directory;
- Load of a key file from the location specified in the configuration file. If the key file is not found, loading of **Dr.Web Daemon** terminates;
- If the **`User`** parameter is specified, **Dr.Web Daemon** attempts to change its privileges;
- Load of **Dr.Web Engine** (**`drweb32.dll`**). If **Dr.Web Engine** is damaged or not found (because of errors in the configuration file), initialization of **Dr.Web Daemon** terminates;
- Load of virus databases in arbitrary sequence from the location specified in the configuration file. If virus databases are damaged or absent, initialization of **Dr.Web Daemon** proceeds;
- **Dr.Web Daemon** enters daemon mode, so all information about initialization problems cannot be output to the console and is logged to the log file;
- Creation of a socket for interaction between **Dr.Web Daemon** and other **Dr.Web for Novell Storage Services** modules. When TCP-sockets are used, there can be several connections (loading continues if at least one connection is established). When a UNIX socket is used, **Dr.Web Daemon** user account must have appropriate privileges to read and write from the directory of this socket. User accounts for modules must have execution access to the directory and write and read access to the socket file. Users do not have write permission for the default socket directory (`/var/run/`). If the **`User`** parameter is specified, adjust the **`Socket`** parameter and provide alternative path to the socket file. If creation of the UNIX socket was unsuccessful, initialization of **Dr.Web Daemon** terminates;
- Creation of a PID file with **Dr.Web Daemon** PID information and transport addresses. User account under which **Dr.Web Daemon** is started must have appropriate privileges to write to the directory of the PID file. Users do not have write permission for the default socket directory (`/var/run/`). So, if the **`User`** parameter is specified, adjust the **`PidFile`** parameter and provide alternative path to the PID file. If creation of the PID file was unsuccessful, initialization of **Dr.Web Daemon** terminates.

Dr.Web Daemon Testing and Diagnostics

If no problems occurred during initialization, **Dr.Web Daemon** is ready to use. To ensure that the daemon is initialized correctly, use the following command:

```
$ netstat -a
```

and check whether required sockets are 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
. . .
```

Missing of the required sockets in the list indicates problems with **Dr.Web Daemon** initialization.

To perform a functional test and obtain service information, use **Dr.Web Daemon console client** (**drwebdc**).

TCP sockets:

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

Unix socket:

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

Report, similar to the following example, is output to the console:

```
- 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 the report was not output, run extended diagnostics.

For TCP socket:

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

For UNIX socket:

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

More detailed report can help to 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 the special **eicar.com** program included in the installation package. Use any text editor 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
```

The following result are output:

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

If the results were not output, check **Dr.Web Daemon** log file to see whether the file was scanned. If the file was not scanned, run extended diagnostic (see above).

If file was scanned successfully, **Dr.Web Daemon** is fully operational.

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

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

Scan Modes

Dr.Web Daemon has two scan modes:

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

In the **remote scan mode**, client sends data to be scanned to **Dr.Web Daemon** through a socket. **Dr.Web Daemon** can scan both anonymous memory and memory mapped objects with only one difference - in logging. This mode enables scanning of files without read access but is less efficient than the local scan mode.

Local scan mode is easier to use and provides better performance since client sends to **Dr.Web Daemon** only a file path instead of the file. For the reason that clients can be located on different computers, the path must be specified in relation to the actual location of **Dr.Web Daemon**.

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

Note that to enable correct operation of **Dr.Web Daemon** as a part of **Dr.Web for Novell Storage Services**, the component must be started under the `root` superuser account.

If required, name of the user with whose privileges **Dr.Web Daemon** must run is set as the **`User`** parameter value in **Dr.Web Daemon** settings. In addition, you can configure user and their group used on module startup. For that purpose, edit mmc-file of **Dr.Web Monitor** if it is used for management of **Dr.Web for Novell Storage Services** components.

Processed Signals

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

- `SIGHUP` – reload the configuration file;
- `SIGTERM` – correct termination of **Dr.Web Daemon**;
- `SIGKILL` – force termination of **Dr.Web Daemon** (if any problem occurs);
- `SIGUSR1` – save process pool statistics to the log file.

Please note that `SIGUSR1` signal must be sent to its parent process only, because child processes are terminated after receiving of `SIGUSR1`.

Log Files and Statistics

Daemon Log

Since **Dr.Web Daemon** is a resident program, information on its operation can be obtained only from a log file. Log file contains details on processing of all scanning request sent to **Dr.Web Daemon**. You can specify the log file location in a value of the **`LogFileName`** parameter.

Dr.Web Daemon can log information to different files depending on a client that sent the request. You can specify different log files for every **Dr.Web** clients (for example, **Dr.Web for Novell Storage Services**) in the **`ClientsLogs`** parameter value.

Regardless of the **`ClientsLogs`** parameter, if **Dr.Web Daemon** recognizes its client, scanning results will marked with a prefix indicating the client. The following prefixes are available:

- `<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**.

In the **FreeBSD** operating system, **`syslog`** service can intercept information output by **Dr.Web Daemon** to the console. In this case, the information is logged character-by-character. That occurs when the logging level is set to `*.info` in the **`syslog`** configuration file (`syslog.conf`).

Statistics on process pool

Statistics on pool used for processing scanning request is output to the log file upon receipt of `SIGUSR1` signal (the signal must be sent only to parent process, as if a child process receives `SIGUSR1`, it terminates).

Output of statistics on process pool is regulated by the **`stat`** value (`yes` or `no`), specified for the

ProcessesPool parameter. Collected statistics is not aggregated. Each time the saved record contains statistics on the pool state between previous and current moment of saving.

Example of pool statistics output record:

```
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)` – displays if limits on number of processes in the pool are determined automatically;
- `freetime` – maximum idle time for a process in the pool;
- `busy max` – maximum number of simultaneously used processes, `avg` - average number of simultaneously used processes;
- `requests for new process` – number of requests for new process creation (frequency of requests per second is displayed in parenthesis);
- `creating fails` – number of failed attempts to create a new process (failures usually occur when the system is running low on resources);
- `max processing time` – maximum time for processing a single scanning request;
- `avg` – average time for processing a single scanning request;
- `curr` – number of all current processes in the pool;
- `busy` – number of currently used processes in the pool.

Configuration

Dr.Web Daemon can be run with default settings, but you can configure it according to your specific requirements. **Daemon** settings are stored in the `[Daemon]` section of the configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory. To use another configuration file, specify the full path to it as a 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 virus databases. This parameter is also used by **Dr.Web Updater**. Multiple values are allowed (separated by commas). By default, virus databases files has the `.vdb` extension Default value: **VirusBase** = %var_dir/bases/*.vdb
UpdatePath = {path to directory}	Directory to store updates. The parameter is mandatory. Default value: **UpdatePath** = %var_dir/updates/

TempPath = {path to directory}	Directory where the **Dr.Web Engine** anti-virus 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). By default, a key file has the .key extension. Please note that **Dr.Web Daemon** and **Dr.Web Scanner** can have different license key files. In this case, change the value of this parameter correspondingly. The parameter value can be set several times to specify several license key files. In this case, **Dr.Web Daemon** tries to combine all license permissions from all available license key files. 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 **Dr.Web Daemon**. With Yes value specified **Dr.Web Daemon** runs as a foreground process. This parameter can be used for certain monitoring utilities (for example, **Dr.Web Monitor**). Default value: **RunForeground** = No
User = {text value}	User under which **Dr.Web Daemon** operates. It is strongly recommended to create a separate **drweb** user account, which will be used by **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 configure. 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 the **Socket** parameter) or port number (if TCP socket is enabled by the **Socket** parameter). If more than one **Socket** parameter is specified, this file contains 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** busy flag is stored. This file is created by a **Dr.Web Daemon** child process upon

	receipt of the scan command and is removed after successful command execution. Filenames created by each **Dr.Web Daemon** child process are appended by a dot and ASCII representation of the PID (for example, `/var/run/drwebd.bsy.123456`). Default value: **BusyFile** = %var_dir/run/drwebd.bsy
ProcessesPool = {process pool settings}	Settings of dynamic process pool. At first, specify the number of processes in the pool: • `auto` – number of processes is set automatically depending on system load; • `N` – nonnegative integer. Pool will have at least `N` active processes, additional processes will be created if necessary; • `N-M` – positive integer, `M>=N`. The 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 specify optional secondary parameters: • **timeout** = {time in seconds} – timeout for closing an inactive process. This parameter does not affect the first `N` processes which wait for requests indefinitely. • **stat** = {yes|no} – statistics on 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 time to wait for a running process to stop. Default value: **ProcessesPool** = auto,**timeout** = 120, **stat** = no, **stop_timeout** = 1
OnlyKey = {logical}	Enables receiving only a license key file from **Dr.Web Agent**, without configuration. At that, **Dr.Web Scanner** uses the local configuration file. If the value is set to `No` and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Daemon** sends operational statistics to **Dr.Web Agent** (information is sent after scanning of every file). Default value: **OnlyKey** = No
ControlAgent = {address}	**Dr.Web Agent** socket address. **Example**: **ControlAgent** = inet:4040@127.0.0.1,local:%var_dir/ipc/.agent **Dr.Web Daemon** receives from **Dr.Web Agent** a license key file (and configuration if **OnlyKey** = No. Moreover, in this case the socket is used for sending statistics on **Dr.Web Daemon** operation to **Dr.Web Agent**). Default value: **ControlAgent** = local:%var_dir/ipc/.agent
MailCommand = {string}	Shell command used by **Dr.Web Daemon** and **Dr.Web Updater** for sending notifications on new updates to the user

	(administrator) via email. If the period before the key file (or one of the key files) expiration is less than the period specified by the **NotifyPeriod** parameter, **Dr.Web Daemon** starts sending notifications upon every system startup, restart or reboot. Default value: **MailCommand** = "/usr/sbin/sendmail -i -bm -f drweb -- root"
NotifyPeriod = {numerical value}	This parameter value specifies the period (in days) before license key expiration date when **Dr.Web Daemon** starts prompting a user to renew the license. If the parameter value is set to 0, **Dr.Web Daemon** starts sending out notifications immediately after the key file expires. Default value: **NotifyPeriod** = 14
NotifyFile = {path to file}	Path to the file with a timestamp of the last license expiration notification. Default value: **NotifyFile** = %var_dir/.notify
NotifyType = {Ever | Everyday | Once}	Frequency of sending license expiration notifications. • Once – notification is sent only once. • Everyday – notification is sent daily. • Ever – notification is sent upon every **Dr.Web Daemon** restart and every database update. Default value: **NotifyType** = Ever
FileTimeout = {numerical value}	Maximum time (in seconds) allowed for **Dr.Web Daemon** to perform scanning of one file. If the parameter value is set to 0, time to scan of one file is unlimited. Default value: **FileTimeout** = 30
StopOnFirstInfected = {logical}	Enables or disables interruption of file scanning upon detection of the first virus. If the value is set to yes, it can significantly reduce mail server load and scan time. 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}	Types of files to be checked "by type", that is, when the **ScanFiles** parameter value (described below) is set to ByType.

	"*" 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 on files of unknown types Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Scan only files with extensions specified in the **FileTypes** parameter (the ByType value) or all files (the All value). This parameter can have the ByType value only in the **local scan** mode (in other modes, only the All value can be set). In mailboxes, all files are always checked (regardless of the **ScanFiles** parameter value). Default value: **ScanFiles** = All
CheckArchives = {logical}	Enables or disables checking of files in archives. The following formats are supported: ZIP (WinZip, InfoZIP, etc.), RAR, ARJ, TAR, GZIP, CAB and others. Default value: **CheckArchives** = Yes
CheckEMailFiles = {logical}	Enables or disables checking of email files. Default value: **CheckEMailFiles** = Yes
ExcludePaths = {list of path | file masks}	Masks for files to be skipped during scanning. Default value: **ExcludePaths** = /proc,/sys,/dev
FollowLinks = {logical}	Enables or disables **Dr.Web Daemon** to follow symbolic links during scanning. Default value: **FollowLinks** = No
RenameFilesTo = {mask}	Mask for renaming files when the Rename action is applied. 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 cured files. Default value: **BackupFilesTo** = %var_dir/infected/

LogFileName = {syslog | file name}	Log file name. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. In this case, also specify the **SyslogFacility** and **SyslogPriority** parameter values. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label used by `syslogd` system service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Logging priority (log verbosity level) when `syslogd` system service is used. There are the following levels 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** checks the size of a log file on startup or on receipt of `HUP` signal. If the log file size is greater than `MaxLogSize`value, the log file is overwritten with an empty file and logging starts 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 a log file to be unexpectedly modified on startup. 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. The parameter is not used if **LogFileName** = syslog. Default value: **LogTime** = Yes
LogProcessInfo = {logical}	Enables or disables logging PID of the scanning process and filter address (host name or IP address) from which scanning has been activated. This data is logged before each record. Default value: **LogProcessInfo** = Yes
RecodeNonprintable = {logical}	Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters). 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 the **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable, all non-printable characters are converted to Quoted Printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets a character to replace all 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 commas). **Example**: Socket = inet:3000@127.0.0.1,local:%var_dir/.daemon You can also specify a socket address in the following format: PORT [interfaces] | FILE [access]. For a TCP socket, specify a 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 a socket name (FILE) and access permissions in the octal form. **Example:** **Socket** = %var_dir/.daemon 0660 Number of **Socket** parameter values is not limited. **Dr.Web Daemon** will work with all sockets described correctly.

	To enable connections on all available interfaces, set 3000 0.0.0.0 as a value of 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 the parameter value is set to 0, the time is unlimited. Default value: **SocketTimeout** = 10
ClientsLogs = {string list}	Enables splitting of log files. If during communication with **Dr.Web Daemon** a 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 separated by commas or spaces. If more than six values are set, the configuration file is considered invalid. 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, a message notifying that databases are obsolete is output. If value is set to 0, database obsolescence is not checked. Default value: **MaxBasesObsolescencePeriod** = 24

The following parameters can be used to reduce scanning time in archived files (some objects in archives are not checked). Actions applied to skipped depend on the **ArchiveRestriction** parameter value of the corresponding modules.

MaxCompressionRatio = {numerical value}	Maximum compression ratio, that is a ratio between size of unpacked file and its size within an archive. The parameter can have only natural values. If the ratio exceeds

	the specified value, file will not be extracted and therefore will not be checked. Value of this parameter must be not less than 2. Default value: **MaxCompressionRatio** = 5000
CompressionCheckThreshold = {numerical value}	Minimum size of a file enclosed within an archive (in Kbytes) for which compression ratio check is performed (if such a check is enabled by the **MaxCompressionRatio** parameter). Value of this parameter must be greater than 0. Default value: **CompressionCheckThreshold** = 1024
MaxFileSizeToExtract = {numerical value}	Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped. Default value: **MaxFileSizeToExtract** = 40960
MaxArchiveLevel = {numerical value}	Maximum allowed archive nesting level. If an archive nesting level exceeds the specified value, an archive is not scanned. Default value: **MaxArchiveLevel** = 8
MessagePatternFileName = {path to file}	Path to template for a license expiration message. You can configure output of an expiration message according to your needs. To do this, use the following variables in the template. The specified variables are substituted with the corresponding values: • $EXPIRATIONDAYS — number of days left until license expiration; • $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 is output. Default value: **MessagePatternFileName =** %etc_dir/templates/drwebd/msg.tmpl
MailTo = {email address}	Email address of an administrator where the following information is sent: messages about license expiration, virus databases obsolescence, etc. Default value: **MailTo =**

Dr.Web Command Line Scanner

Command line **Dr.Web Scanner** provides you with detection and neutralization of malware on the local machine. The component is presented by the **drweb** module.

Dr.Web Scanner checks files and boot records specified on its startup. For anti-virus checking and curing, **Dr.Web Scanner** uses **Dr.Web Engine** and virus databases, but does not use the resident module **Dr.Web Daemon** (operation is performed independently of it).

Running Dr.Web Scanner

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

```
$ %bin_dir/drweb
```

If `%bin_dir` directory is added to the `PATH` environment variable, you can run **Dr.Web Scanner** from any directory. 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 latter case, virus scanning can be performed only in those directories, where the user has read access, and infected files will be cured only in directories, where the user has write access (usually it is the user home directory, `$HOME`). There are also 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 the 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 and command line prompt. To scan for viruses or neutralize detected threats, 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 of the parameters is not required, disable it with "-" postfix as described above.

Disabling scan of archives and packed files will significantly decrease an anti-virus protection level, because viruses are often distributed in archives (especially, self-extracting archives) attached to an email message. Office documents (Word, Excel) dispatched within an archive or a container can also pose a threat to security of your computer as they are vulnerable to macro viruses.

When you start **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are performed. To enable these actions, specify the corresponding command line parameter explicitly.

The 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 tries to restore the original state of an infected object. It is possible only if a detected virus is a known virus, and cure instructions for it are available in virus database; even in this case a cure attempt may fail if the infected file is seriously damaged by a virus.

When an infected file is found within an archive, the file is not cured, deleted, moved or renamed. To cure such a file, manually unpack the archive to the separate directory and instruct **Dr.Web Scanner** to check it.

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

The **spr** action instructs **Dr.Web Scanner** to replace a file extension with another one (`*.#??` by default, that is the first extension character is replaced with the "#" character). Enable this parameter for files of other operating systems, detected heuristically as suspicious. Renaming helps to avoid accidental execution of such files in these operating systems and therefore prevents infection.

The **spm** action instructs **Dr.Web Scanner** to move infected or suspicious files to the **Quarantine** directory (`%var_dir/infected/` by default). This option is of insignificant value since infected and suspicious files of other operating systems cannot infect or damage a UNIX system. Moving of suspicious files of a UNIX system may cause system malfunction or failure.

Thus, the following command is recommended 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 the following command:

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

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

Command Line Parameters

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

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

where `<path>` – is either the path (or paths) to scanned directories or mask for checked files. If a path is specified with the following prefix: `disk://<path to device file>` (files of the devices are

located in the /dev directory), **Dr.Web Scanner** checks the boot sector of the corresponding device and cure it, if necessary. The path can start with an optional parameter - path.

When **Dr.Web Scanner** is started only with the <path> argument, without any parameters specified, it scans the specified directory using the default set of parameters (for details, see below).

The following example shows a command to check the user home directory:

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

Once scanning completes, **Dr.Web Scanner** displays all detected threats (infected and suspicious files) in the following format:

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

After that, **Dr.Web Scanner** outputs summary report in the following format:

```
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 separated by slash "/" mean the following: the first number – 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 any text editor and follow the instructions from the file to transform it into eicar.com program.

When you check the program with **Dr.Web Scanner**, the following message must be output:

```
%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 software.

Dr.Web Scanner has numerous command-line parameters. In accordance with UNIX conventions, the parameters are separated from a path by a space character and start with a hyphen ("-"). To get a full list of parameters, run **Dr.Web Scanner** with either -?, -h, or -help parameters.

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

- Scan area parameters
- Diagnostic parameters
- Action parameters
- Interface parameters

Scan Area Parameters

These parameters determine where to perform a virus scan:

Parameter	Description
-path [=] <path>	Sets the path to be scanned. Symbol '=' can be skipped, in this case a path for scanning is separated from the -path parameter by a space. You can specify several paths in one -path parameter (paths will be combined into one list). You can also specify paths without the -path parameter. If in the startup options the <path> parameter is specified with following prefix: disk://<path to device file>,

Parameter	Description
	the boot sector (MBR) of the corresponding device is checked and cured, if necessary. Device file is a special file, located in the `/dev` directory and named as **sdX** or **hdX**, where **X** is a letter of the Latin alphabet (`a`, `b`, `c`, ...). For example: **hda**, **sda**. Thus, to check MBR of disk **sda**, specify the following: `disk:///dev/sda`
`-@[+]<file>`	Instructs to scan objects listed in the specified file. Add a plus '+' if you do not want the file with the list of objects to be deleted when scanning completes. The file can contain paths to directories that must be periodically scanned or list of files to be checked regularly.
`--`	Instructs to read the list of objects for scanning from the standard input stream (`stdin`).
`-sd`	Sets recursive search for files to scan in subfolders.
`-fl`	Instructs to follow symbolic links to both files and folders. Links that cause loops are ignored.
`-mask`	Instructs to ignore filename masks.

Diagnostic Parameters

These parameters determine object types to be scanned 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 to the **-ex** parameter.
-ex	Instructs to scan only files of certain types in the specified paths. The list of file types must 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 to the **-al** parameter.
-ar`[d|m|r][n]`	Instructs to scan files within archives (ARJ, CAB, GZIP, RAR, TAR, ZIP, etc.). An archive is understood to be a tar archive (*.tar) or compressed archive (*.tar.bz2, *.tbz). If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious files in archives. Otherwise, it applies the specified actions to detected threats.
-cn`[d|m|r][n]`	Instructs to scan files within containers (HTML, RTF, PowerPoint). If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious files in containers. Otherwise, it applies the specified actions to detected threats.
-ml`[d|m|r][n]`	Instructs to scan contents of mail files. If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious objects. Otherwise, it applies the specified actions to detected threats.
-upn	Scan executable files packed with LZEXE, DIET, PKLITE, EXEPACK without output of the compression type.
-ha	Enables heuristic analysis to detect 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 extension with '#')
- Add `n` to disable logging 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, the reaction is applied to the whole complex object, and not to the included malicious object only.

Action Parameters

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

Parameter	Description
`-cu[d|m|r]`	Defines an action applied to infected files and boot sectors. If an additional modifier is not specified, **Dr.Web Scanner** cures infected objects and deletes incurable files (unless another action is specified in the `-ic` parameter). Additional modifiers allow to set another action instead of curing, but the new action can be applied only to infected files. In this case, action for incurable files must be set with `-ic` parameter.
`-ic[d|m|r]`	Defines an action applied to incurable files. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-sp[d|m|r]`	Defines an action applied to suspicious files. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-adw[d|m|r|i]`	Defines an action applied to adware. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-dls[d|m|r|i]`	Defines an action applied to dialers. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-jok[d|m|r|i]`	Defines an action applied to joke programs. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-rsk[d|m|r|i]`	Defines an action applied to potentially dangerous programs. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
`-hck[d|m|r|i]`	Defines an actionapplied to hacktools. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.

Additional modifiers indicate actions that is applied in order to avert 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, the action is applied to the whole complex object, 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 on the product and 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 the "clean" object with **Ok**.
-log=[+]<path to file>	Instructs to log **Dr.Web Scanner** operations in the specified file. The file name is required for enabling 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. By default, **Dr.Web Scanner** uses **drweb32.ini** (this configuration file is shared by **Dr.Web Daemon**, **Dr.Web Scanner** and **Dr.Web Updater**). **Dr.Web Scanner** uses parameters specified in the [Scanner] section of this file. The list of the scanner parameters and available values are similar to the those specified in the [Daemon] section.
-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 the central protection mode.
-ni	Disables the use of the configuration file for adjusting scanner settings. **Dr.Web Scanner** is configured via command line parameters.
-ns	Disables interruption of scanning process even upon receipt of interruption signals (SIGINT).
--only-key	On startup, only key file is received from **Dr.Web Agent**.

You can use the hyphen «-» postfix (no space) 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, information on detection of infected or suspicious object is logged, but no action is performed to avert threats.

The -al and -ex parameters have no "negative" form, but specifying one of them cancels actions of the other.

By default (if **Dr.Web Scanner** configuration is not customized and no parameters are specified), **Dr.Web Scanner** is started with the following parameters:

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

Default **Dr.Web Scanner** parameters (including scan of archives, packed files, files of email programs, recursive search, heuristic analysis and others) are sufficient for everyday diagnostics and can be used in most typical cases. You can also use hyphen «-» postfix to disable required parameters (as it is shown above with an example of heuristic analysis).

Disabling scanning of archives and packed files significantly decreases anti-virus protection level, because viruses are often distributed as archives (especially, self-extracting ones) attached to an email message. Office documents are potentially susceptible to infection with macro viruses (e.g., **Word, Excel**) and can also be dispatched via email within archives and containers.

When you run **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are performed. To enable these actions, specify the corresponding command line parameters explicitly.

Configuration

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

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

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

For general principles of the **Dr.Web for Novell Storage Services** configuration files organization, see 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** stores temporary files. It is used for unpacking archives or when the system is low on memory Default value: **TempPath** = /tmp/

LngFileName = {path to file}	Language file location. By default, language files have 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 have 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}	Enables or disables 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 a suspicious file, put it in a password protected archive, include password in the 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). Default value: **ScanPriority** = 0
FilesTypes = {list of file extensions}	File types to be checked "by type", i.e. when the **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}	Notifies about files of unknown types. Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Instructs to scan all files (All value) or only files with the extensions specified in the **FileType** parameter (ByType value).

	The parameter can have the `ByType` value only in the **local scan** mode. In other modes, the value must be set to `All`. All mail fails are scanned regardless of the **`ScanFiles`** parameter value. Default value: **`ScanFiles`** = `All`
`ScanSubDirectories` = `{logical}`	Enables or disables scanning of 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 mail 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}`	Allows or forbids **Dr.Web Scanner** to follow symbolic links during scanning. Default value: **`FollowLinks`** = `No`
`RenameFilesTo` = `{mask}`	Mask for renaming files when the `Rename` action is applied. Default value: **`RenameFilesTo`** = `#??`
`MoveFilesTo` = `{path to directory}`	Path to the **Quarantine** directory. Default value: **`MoveFilesTo`** = `%var_dir/infected/`
`EnableDeleteArchiveAction` =`{logical}`	Enables or disables `Delete` action for complex objects (archives, mailboxes, HTML pages) if they contain infected files. Please note, if the action is enabled, a whole complex object is to be deleted. Use this option carefully! Default value: **`EnableDeleteArchiveAction`** = `No`
`InfectedFiles` = `{action}`	Sets one of the following actions upon detection of an infected file: `Report`, `Cure`, `Delete`, `Move`, `Rename`, `Ignore`. `Delete` and `Move` actions are applied to a whole complex object upon detection of infected files within it. Default value: **`InfectedFiles`** = `Report`

SuspiciousFiles = {action}	Sets one of the following actions upon detection of a suspicious file: Report, Delete, Move, Rename, Ignore. Default value: **SuspiciousFiles** = Report
IncurableFiles = {action}	Sets one of the following actions applied if an infected file cannot be cured (use only if **InfectedFiles** = Cure): Report, Delete, Move, Rename, Ignore. Default value: **IncurableFiles** = Report
ActionAdware = {action}	Sets one of the following actions upon detection of adware: Report, Delete, Move, Rename, Ignore. Default value: **ActionAdware** = Report
ActionDialers = {action}	Sets one of the following actions upon detection of a dialer program: Report, Delete, Move, Rename, Ignore. Default value: **ActionDialers** = Report
ActionJokes = {action}	Sets one of the following actions upon detection of a joke program: Report, Delete, Move, Rename, Ignore. Default value: **ActionJokes** = Report
ActionRiskware = {action}	Sets one of the following actions upon detection of a potentially dangerous program: Report, Delete, Move, Rename, Ignore. Default value: **ActionRiskware** = Report
ActionHacktools = {action}	Sets one of the following actions upon detection of a hacktool: Report, Delete, Move, Rename, Ignore. Default value: **ActionHacktools** = Report
ActionInfectedMail = {action}	Sets one of the following actions upon detection of an infected file in a mailbox: Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedMail** = Report
ActionInfectedArchive = {action}	Sets one of the following actions upon detection of an infected file in an archive (ZIP, TAR, RAR, etc.): Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedArchive** = Report

ActionInfectedContainer = {action}	Sets one of the following actions upon detection of an infected file in a container (OLE, HTML, PowerPoint, etc.): Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedContainer** = Report
	Logging parameters:
LogFileName = {syslog | file name}	Log file name. You can specify syslog as a log file name to use syslogd system service for logging. In this case you must also specify the **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}	Log verbosity level when syslogd system service is used. The following levels are allowed: • Error • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
LimitLog = {logical}	Enables or disables limit of log file size (if **LogFileName** value is not set to syslog). With this parameter enabled, **Dr.Web Scanner** checks log file size on startup. If log file size exceeds the **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. If this parameter value is set to 0, log file size is not checked. 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
LogStatistics = {logical}	Enables or disables logging of scan statistics. Default value: **LogStatistics** = Yes
RecodeNonprintable = {logical}	Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters). 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 the **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable, all non-printable characters are converted to the Quoted Printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets character for replacing non-printable characters if **RecodeMode** = Replace. Default value: **RecodeChar** = "?"

The following parameters can be used to reduce time of scanning archives (by skipping some objects in an archive).

MaxCompressionRatio = {numerical value}	Maximum compression ratio, that is ratio between size of unpacked file and its size within an archive. If a ratio exceeds the specified value, the file will not be extracted and therefore will not be checked. An email message with such an archive is considered as a "mail bomb". Parameter can have only natural values. If the value is set to 0, compression ratio will not be checked Default value: **MaxCompressionRatio** = 5000
CompressionCheckThreshold = {numerical value}	Minimum size of a file enclosed within an archive, in Kbytes. If a file size is less than the specified value, the compression ratio will not be checked (if such a check is enabled by the **MaxCompressionRatio** parameter).

	Default value: **CompressionCheckThreshold** = 1024
MaxFileSizeToExtract = {numerical value}	Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped. An email message with such a file is considered as a "mail bomb". Default value: **MaxFileSizeToExtract** = 500000
MaxArchiveLevel = {numerical value}	Maximum archive nesting level. If an archive nesting level exceeds the specified value, the archive is skipped. An email message with such a file is considered as a "mail bomb". If the value is set to 0, archive nesting level will not be checked Default value: **MaxArchiveLevel** = 8
MaximumMemoryAllocationSize = {numerical value}	Maximum size of the memory (in Mbytes) that can be used by **Dr.Web Scanner** to check one file. If the value is set to 0, memory allocation is not limited. Default value: **MaximumMemoryAllocationSize** = 0
ScannerScanTimeout = {numerical value}	Maximum time period allowed for scanning one file (in seconds). If the value is set to 0, scanning time is not limited. Default value: **ScannerScanTimeout** = 0
MaxBasesObsolescencePeriod = {numerical value}	Maximum time (in hours) after last update when virus databases are considered as up-to-date. Upon the expiration of this time period, notification displays informing that the databases are obsolete. If the value is set to 0, database actuality 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 a license key file and configuration from **Dr.Web Agent**. (if **OnlyKey** = No). Default value: **ControlAgent** = local:%var_dir/ipc/.agent
OnlyKey = {logical}	Enables receiving only a license key file from **Dr.Web Agent**, without configuration. At that, **Dr.Web Scanner** uses the local configuration file. If the value is set to No and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Agent** also receives statistics on **Dr.Web Scanner** operation (information is sent after scanning of each file).

	Default value: **OnlyKey** = No

Exit Codes

When the scan task ends, **Dr.Web Scanner** returns an exit code which determines result of scanning.

The exit code is always constructed as an combination (sum) of codes that are related to the corresponding events of scanning process. The possible events and related codes are following:

Code	Event
1	Known virus detected
2	Modification of known virus detected
4	Suspicious object found
8	Known virus detected in archive, mailbox or other container
16	Modification of known virus detected in archive, mailbox or other container
32	Suspicious file found in archive, mailbox or other container
64	At least one infected object succesfully cured
128	At least one infected or suspicious file deleted/renamed/moved

The actual value returned by **Dr.Web Scanner** is equal to the sum of codes for the events that occurred during scanning. Obviously, the sum can be easily decomposed into separate event codes. For example, return code 9 = 1 + 8 means that known viruses were detected, including viruses in archives, mail archives or containers; curing and others actions were not executed; no other threat events occurred during scanning.

If no threat events occurred during scanning, **Dr.Web Scanner** returns the exit code 0.

Dr.Web Scanner has one feature: in some cases, when no threats were found during scanning, it can return the exit code 128 instead of exit code 0. This case is similar to the case "no threats found" (exit code 0).