



Dr.WEB®

Anti-virus + Anti-spam
for UNIX mail servers

Defend what you create

Administrator Manual

© Doctor Web, 2014. All rights reserved

This document is the property of Doctor Web. No part of this document may be reproduced, published or transmitted in any form or by any means for any purpose other than the purchaser's personal use without proper attribution.

TRADEMARKS

Dr.Web, SpIDer Mail, SpIDer Guard, CureIt!, CureNet!, Dr.Web AV-Desk and the Dr.WEB logo are trademarks and registered trademarks of Doctor Web in Russia and/or other countries. Other trademarks, registered trademarks and company names used in this document are property of their respective owners.

DISCLAIMER

In no event shall Doctor Web and its resellers or distributors be liable for errors or omissions, or any loss of profit or any other damage caused or alleged to be caused directly or indirectly by this document, the use of or inability to use information contained in this document.

**Anti-virus Dr.Web for UNIX mail servers
Version 6.0.2
Administrator Manual
01.12.2014**

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	12
Terms and Abbreviations	15
System Requirements	17
Compatibility with Linux Distributions	18
Package File Location	19
Configuration Files	20
Logging	23
Allowed Actions	24
Installation and Deinstallation	26
Installation from Distribution Package for UNIX Systems	26
Using GUI Installer	29
Using Console Installer	34
Removing Distribution Package for UNIX Systems	37
Using GUI Uninstaller	38
Using Console Uninstaller	40
Installing from Native Packages	41
Configuration Scripts	47
Starting Dr.Web for UNIX mail servers	48
For Linux and Solaris OS	48
For FreeBSD OS	49
Configuring SELinux Security Policies	50
Registration Procedure	53
Dr.Web Command Line Scanner	55
Running Dr.Web Scanner	55
Command Line Parameters	56
Configuration	61
Exit Codes	68
Dr.Web Daemon	69
Command-Line Parameters	69
Running Dr.Web Daemon	70
Dr.Web Daemon Testing and Diagnostics	70
Scan Modes	72
Processed Signals	73



Log Files and Statistics	73
Configuration	74
Dr.Web Updater	84
Updating Anti-Virus and Virus Databases	84
Cron Configuration	85
Command Line Parameters	86
Blocking Updates for Selected Components	86
Restoring Components	87
Configuration	87
Updating Procedure	91
Dr.Web Agent	92
Operation Mode	93
Command Line Parameters	94
Configuration File	95
[Logging] Section	96
[Agent] Section	96
[Server] Section	97
[EnterpriseMode] Section	98
[StandaloneMode] Section	99
[Update] Section	100
Running Dr.Web Agent	100
Interaction with Other Suite Components	101
Integration with Dr.Web Enterprise Security Suite	102
Configuring Components to Run in Enterprise Mode	102
Automatic Creation of New Account by ES Server	103
Manual Creation of New Account by Administrator	103
Configuring Components via Dr.Web Control Center (embedded in Enterprise Security Suite)	103
Export of Existing Configuration to ES Server	104
Starting the System	104
Integration with Dr.Web ESS 10	104
Gathering Virus Statistics	105
Dr.Web Monitor	109
Operation Mode	109
Command Line Parameters	110
Configuration File	110
[Logging] Section	111



[Monitor] Section	111
Running Dr.Web Monitor	114
Interaction with Other Suite Components	114
Dr.Web MailD	116
Message Processing	119
Used Modules	122
Command Line Parameters	123
Processed Signals	127
Logging	128
Internal Statistics	128
RFC Standards	131
Adjustment and Startup	132
Dr.Web MailD Configuration Files	132
Special Parameter Types	133
Lookup	135
Lookup Usage Examples	138
Lookup Usage Restrictions. LookupLite Type	139
Storage Data Type	139
Sections of Main Configuration File	140
General Parameters	142
[General] Section	142
[Mail] Section	143
[MailBase] Section	149
[Notifier] Section	150
[Quarantine] Section	152
[Filters] Section	156
[Rule] Section	159
[Rules] Section	160
[Stat] Section	161
[Reports] Section	164
[Logging] Section	165
Using SASL	166
[SASL] Section	166
[Cyrus-SASL] Section	167
MTA Connections	168
[Receiver] Section	169



[Sender] Section	181
[Courier] Section	187
[CgpReceiver] Section	188
[CgpSender] Section	189
[Milter] Section	190
[Qmail] Section	192
[IMAP] Section	193
[POP3] Section	196
Data Sources	198
[LDAP] Section	198
[Oracle] Section	201
[ODBC] Section	203
[SQLite] Section	204
[Firebird] Section	205
[PostgreSQL] Section	206
[MySQL] Section	207
[CDB] Section	209
[Berkeley] Section	210
Internal Proxy	211
[ProxyClient] Section	211
[ProxyServer] Section	212
Statistics	213
Exporting Statistics	214
Quarantine	214
Using DBI	215
Using Control Email Messages	216
Migrating to New Quarantine Version	216
Interactive Management	216
General Management Commands	217
User, Group, and Alias Management	219
Commands for User Management	222
Commands for Alias Management	225
Commands for Group Management	225
Working with Quarantine	227
Commands for Quarantine Management	227
Receiving Statistics	230



Commands for Working with Statistics	230
Statistics Output	232
Examples	233
Checking Notification Generation	234
Dr.Web MailD Utilities	236
drweb-qcontrol: Quarantine Management	236
drweb-lookup: Lookup Validation	240
drweb-inject: Sending Mail	241
Message Processing Rules	242
Rule Format	243
Special Cases	253
Error Handling and Rule Validation	258
Unified Score	259
Reputation IP Filter	259
Simultaneous Use of Several Receiver/Sender Components	263
Optimizing Operation and Use of System Resources	265
Using Internal Proxy	270
Integration with Cyrus SASL	274
Notification Templates	276
Used Macros	279
Control Constructions	283
Template Example	285
Language Files	289
Plug-Ins	291
Drweb Anti-Virus Plug-In	291
Connecting Drweb Plug-In	291
Setting Drweb Plug-In	292
Examples	299
Vaderetro Anti-Spam Plug-In	300
Installing Vaderetro Plug-In	301
Setting Vaderetro Plug-In	302
Examples	308
Dr.Web HeadersFilter Plug-In	308
Connecting Headersfilter Plug-In	308
Setting Headersfilter Plug-In	309
Examples	311



Dr.Web Modifier Plug-In	311
Connecting Modifier Plug-In	322
Setting Modifier Plug-In	323
Examples	324
Work with String Values	325
Integration with Mail Transfer Systems	326
Working in SMTP Proxy Mode	328
SMTP Callback Mode	329
Working with POP3/IMAP Mail Clients	330
Integration with CommuniGate Pro	332
Configuring CommuniGate Pro	332
Configuring Dr.Web MailD	333
Operation Principles	334
Known Issues	335
Integration with Sendmail	335
Configuring Sendmail	336
Configuring Dr.Web MailD	338
Known Issues	338
Integration with Mail Postfix	339
Main Operation Principles	339
Operation in After-Queue and Before-Queue Modes	339
Operation Using Milter Protocol	340
Configuring Mail Postfix	340
Configuring Operation in After-Queue Mode	340
Configuring Operation Using Milter Protocol	341
Configuring Dr.Web MailD	342
Integration with Exim	343
Configuring Exim	343
Connecting to Exim Using Special Transport	344
Connecting to Exim Using Local_Scan Function	345
Configuring Dr.Web MailD	347
Known Issues	348
Integration with Qmail	348
Configuring Qmail	348
Configuring Dr.Web MailD	349
Known Issues	350



Integration with ZMailer	351
Content Filter at SMTP Session Stage	351
Content Filter at Routing Stage	352
Additional Settings	352
Integration with Courier	352
Configuring Courier	353
Configuring Dr.Web MailD	353
Known Issues	354
Dr.Web Console for UNIX mail servers	357
Installation	357
Basic Configuration	359
User Interface	361
Quarantine	361
Toolbar	362
Filter Panel	363
List of Messages	364
Navigation Pane	366
Configuration	366
Basic Settings Tab	368
Quarantine Tab	369
Plug-ins Tab	369
Anti-Spam Tab	370
Headers Filter Tab	371
Anti-Virus Tab	373
Modifier Tab	374
Rules Tab	376
Engine Tab	379
Reports Tab	382
Mail Receiving Tab	383
Sending Mail Tab	385
IMAP Tab	386
POP3 Tab	387
Proxy Tab	388
Templates	388
Running in Enterprise Mode	389
Configuring User Permissions	390



Configuring Workstation	391
Types of Administrator Accounts	393
Contacts	394



Introduction

This Manual describes the following **Dr.Web®** solutions for mail processing and filtering in UNIX® based systems:

- **Anti-virus + Anti-spam Dr.Web for UNIX mail servers;**
- **Anti-spam Dr.Web for UNIX mail servers;**
- **Anti-virus Dr.Web for UNIX mail servers;**
- **Anti-virus + Anti-spam Dr.Web for UNIX mail gateways;**
- **Anti-spam Dr.Web for UNIX mail gateways;**
- **Anti-virus Dr.Web for UNIX mail gateways.**

In fact, all these solutions differ from each other only by combination of installed modules and plug-ins. Hereinafter all of them will be referred to as **Dr.Web for UNIX mail servers**. Depending on the set of installed modules, interaction with different mail transfer systems can be established, successful mail protection from viruses and spam can be achieved and also the suite can operate as a mail gateway.

Each solution is also presented in three variations for major UNIX based operating systems ("*UNIX systems*" hereinafter): **Linux**, **FreeBSD** and **Solaris x86**.

As far as all these solutions for UNIX systems differ from each other only slightly, all of them will be referred to as **Dr.Web for UNIX mail servers**. Critical differences are described in the corresponding chapters and paragraphs.

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

Filtration of email correspondence in UNIX systems has the following features:

- Monitoring of all incoming SMTP traffic to provide virus detection and neutralization.
In most cases, viruses are not directly aimed at UNIX systems. For example, through electronic mail, ordinary **Windows** viruses are distributed, including macro-viruses for **Word**, **Excel** and other office applications.
- Filtration of spam and other unsolicited messages.

Dr.Web for UNIX mail servers performs both tasks mentioned above.

A range of problems to be solved by **Dr.Web for UNIX mail servers** is limited only by the set of installed components (receivers of incoming messages and senders of outgoing messages) and plug-ins (special libraries responsible for direct processing of messages).

If necessary, a user can extend function set of the solution by adding new modules created in C or C++ programming languages. That is possible due to the use of unified interface (API). To access the API, new client modules must include special header files and libraries (SDK). The following SDK are available:

- SDK for development of new modules that perform functions of **Receiver/Sender** components and provide support to new MTAs.
- SDK for development of new plug-ins for mail processing.

Use of the API for new modules allows to connect them to the product as the standard modules and plug-ins.

Both of the SDK include header files to be used, examples (with the source code) and documentation. These SDK are not provided with the product but they can be downloaded from the repository of **Doctor Web** company. If you encounter any problem, please contact [technical support](#).



Dr.Web for UNIX mail servers 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;
- **Dr.Web MailD** - a component that provides processing of the mail traffic and integration of other **Dr.Web** components with the **Sendmail**, **Postfix**, **Courier**, **Qmail**, **CommuniGate Pro**, **ZMailer** and **Exim** mail transfer systems. In addition to that, this component can perform functions of a mail proxy, that is direct checking of SMTP/LMTP mail traffic. **Dr.Web MailD** can also operate as a part of anti-virus network under control of **Dr.Web Enterprise Security Suite**. The component is delivered with a set of plug-ins which are used for mail processing, for example:
 - Anti-virus plug-in **Drweb** (using **Dr.Web Daemon**).
 - Anti-spam plug-in **Vaderetro**.
- **Dr.Web Console for UNIX mail servers** – web management interface, a **Webmin** built-in module, used for **Dr.Web for UNIX mail servers** management and configuration via the web interface from any browser.

The following picture shows the structure of **Dr.Web for UNIX mail servers** and its components.

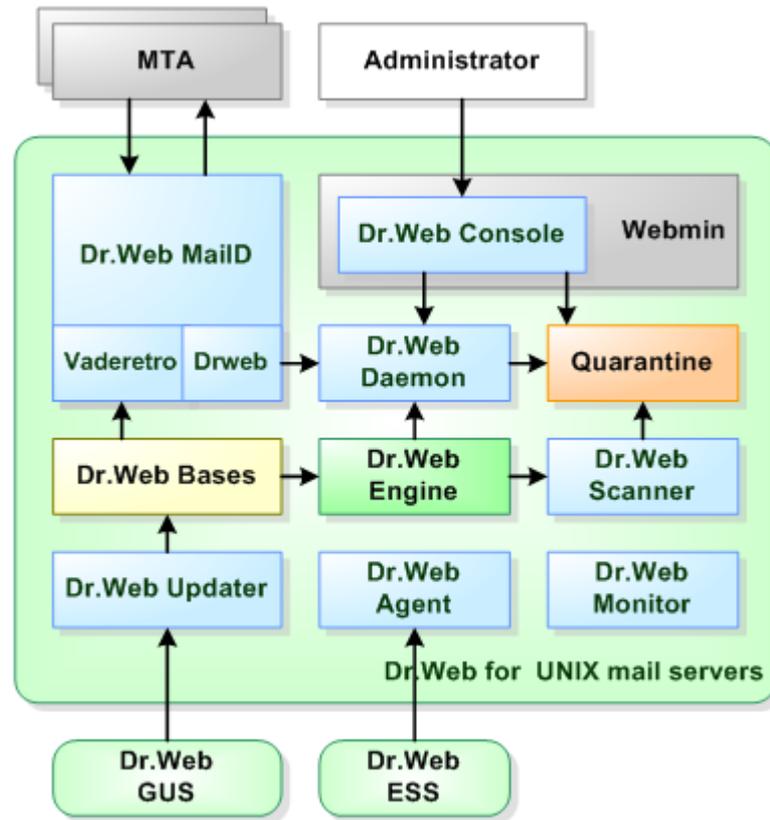


Figure 1. Structure of Dr.Web for UNIX mail servers and its components

The present manual provides information on setup, configuration, and usage of **Dr.Web for UNIX mail servers**, that is:

- General product description
- Installation of **Dr.Web for UNIX mail servers**
- Running **Dr.Web for UNIX mail servers**
- 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**
- Configuration of **Dr.Web for UNIX mail servers** solution.

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.
Green and bold	Names of Doctor Web products and components.
<u>Green and underlined</u>	Hyperlinks to topics and web pages.
Monospace	Code examples, input to the command line and application output.
<i>Italics</i>	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`. Depending on the OS, these symbols refer to the following directories:

for Linux and Solaris:

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

for FreeBSD:

```
%bin_dir = /usr/local/drweb/  
%etc_dir = /usr/local/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



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

The following conventions are used in the chapters about **Dr.Web MailD** and **Dr.Web Console for UNIX mail servers**:

Abbreviation	Description
A record	DNS record about correspondence between hostmane and IP address
API	Application Programming Interface
CGI	Common Gateway Interface
CTE	Content Transfer Encoding, i.e. standard of MIME object encoding (7-bit or 8-bit encoding) for email messages
DB	Database
DBMS	DataBase Management System
DNSBL	DNS blacklist (DNS blocklist) – a list of IP addresses published through the DNS. Most often used to publish the addresses of computers or networks linked to spamming
DSN	Delivery Status Notification
DSN	Data Source Name in ODBC
FQDN	Fully Qualified Domain Name, i.e. a domain name that can be interpreted unambiguously. The name includes names of all parent domains in the DNS hierarchy.
IMAP	Internet Message Access Protocol
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
LMP	Local Mail Transfer Protocol
MIME	Multipurpose Internet Mail Extensions
MDA	Mail Delivery Agent – agent which is responsible for mail delivery (usually an integral part of a mail server)
MTA	Mail Transfer Agent – mail server or its integral part which is responsible for mail receipt and mail transfer
MUA	Mail User Agent
MX record	DNS record about IP addresses of Mail agents into domain
ODBC	Open Database Connectivity API
POP3	Post Office Protocol Version 3
SASL	Simple Authentication and Security Layer framework
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol



System Requirements

Dr.Web for UNIX mail servers is compatible with

- **Linux** distributions that meet requirements listed in [Compatibility with Linux Distributions](#);
- **FreeBSD** version 6.x and higher for Intel x86 and amd64 platform;
- **Solaris** version 10 for Intel x86 and amd64 platform.



Used platform must be fully compatible with x86 processor architecture in 32-bit or 64-bit modes. 64-bit systems must support 32-bit applications.

The products, operating in **FreeBSD** 6.x, cannot be [integrated](#) with **Dr.Web ESS** 10.

For example:

To enable support for 32-bit applications in systems based on **Debian/Ubuntu Linux** the `libc6-i386` library must be installed, in systems based on **ALT Linux** - the `i586-glibc-core` library.

For successful operation of **Dr.Web for UNIX mail servers**, it is required to:

- Install and run **Dr.Web Daemon** and anti-virus **Dr.Web Engine** version 6.0.2 or later.
- Installed **Perl** 5.8.0 or later for **Dr.Web Updater**.

Dr.Web for UNIX mail servers hardware requirements are the same as requirements for the command line interface of the compatible operating system.

Installation requires 190 megabytes.

GUI installer of **Dr.Web for UNIX mail servers** requires **X Window System**. Execution of interactive configuration script in graphical mode requires `xterm` or `xvt` terminal emulators.

In addition to that, the following packages must be installed in your system:

- `base64`
- `unzip`
- `cron`

For successful installation of **Dr.Web for UNIX mail servers** in **FreeBSD** OS (version later than 8.0), the `compat7x` library is required.

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



Compatibility with Linux Distributions

Dr.Web for UNIX mail servers solution is compatible with x86 and x86-64 **Linux** distributions.

Requirements for kernel versions and glibc library depend on the type of the installation package:

- Universal package for UNIX systems (Linux x86):
 - **kernel** version 2.4.x, **glibc** version 2.2 (not recommended) and later,
OR
 - **kernel** version 2.6.x, **glibc** version 2.3 and later;
- Universal package for UNIX systems (Linux x86-64):
 - **kernel** version 2.6.x, **glibc** version 2.3 (recommended) and later;
- Native RPM distribution packages (rpm-apt, urpmi, yum, zypper):
 - **kernel** version 2.6.18 and later, **glibc** version 2.5 and later;
- Native DEB distribution packages (apt):
 - **kernel** version 2.6.26 and later, **glibc** version 2.7 and later.

Performance of **Dr.Web for UNIX mail servers** was tested on the following distributions:

Linux distribution	Versions	
	32-bit	64-bit
ALT Linux	4.0 – 5.0 CPT 6.0 (ru)	5.0 CPT 6.0 (ru)
Arch Linux	–	all
ASPLinux	12.0 – 14.0	–
Debian	3.1 – 6.0	4.0 – 6.0
Fedora	–	14.0
Gentoo	all	
Mandriva Linux	higher than 2009, CS4	2010.x
Mandrake	10.x	10.x
openSUSE	10.3 – 11.0	10.3 – 11.0
PCLinux	2010	2010
RedHat Enterprise Linux (RHEL)	4.0 – 6.0	5.0 – 6.0
Suse Linux Enterprise Server	9.0 – 11.0	10.0 – 11.0
Ubuntu	7.04 – 11.04	7.04 – 11.04

Compatibility with MSVS OS

Dr.Web for UNIX mail servers is compatible with the following versions of **MSVS** OS:

- **MSVS** 3.0 80001-12 (rev. 0, 1, 2, 3);
- **MSVS** 3.0 80001-14 (rev. 0, 1, 2);
- **MSVS** 3.0 80001-08;
- **MSVS** 3.0 80001-16;
- **MSVS** 3.0 FSTEK.

Other **Linux** distributions that meet the requirements mentioned above are also supported (but they were not tested). If you encounter any compatibility problems with the used **Linux** distribution, please



contact technical support at <http://support.drweb.com/request/>.

Package File Location

Dr.Web for UNIX mail servers 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 UNIX mail servers** 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 UNIX mail servers** component operation, for example:
 - `ru_scanner.dwl` - file of **Dr.Web Scanner** language resources.
- `%bin_dir/scripts/`,
`%bin_dir/maild/scripts/` - directories with additional scripts, **Dr.Web for UNIX mail servers** autoconfiguration script, migration script for transfer of configuration from older **Dr.Web** versions.
- `%bin_dir/web/` - **Dr.Web for UNIX mail servers** web interface module for connection to **Webmin**.
- `%etc_dir/` - directory with **Dr.Web for UNIX mail servers** 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/maild/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 UNIX mail servers** settings.
- `%var_dir/lib/` - anti-virus engine implemented as a loadable library (`drweb32.dll`).

*) Directory of the `enable` files depends on **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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:

<p>[Status in the Mail Processing Rules]</p> <p>ParameterName = {Parameter type Possible values}</p>	<p>Description</p> <p>{Whether more than one value is possible}</p> <p>{Special remarks}</p> <p>{Important remarks}</p> <hr/> <p>Default value:</p> <p>ParameterName = {value nothing}</p>
---	---

Status can be designated by one of following icons:

- R** The parameter can be used in the `SETTINGS` part of [Mail Processing Rules](#). The parameter is used to temporarily change its value while processing an email message which corresponds the conditional part of the rule.
- A** The parameter, when used in [Mail Processing Rules](#), is additive, meaning that if more than one rule is true for the email message, the parameter value results from joining all values of the rules.
- C** The parameter, when used in [Mail Processing Rules](#), supports cloning of email messages. That is, if the message has several recipients and for different recipients there are rules with different parameter values, the message is to be cloned (according to the number of the recipients). The parameter value is defined separately for each copy by the rule which is true for it.

If a parameter Status is not specified in its description, this parameter cannot be used in [Mail Processing Rules](#).

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 UNIX mail servers** 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



- **permissions** — 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 UNIX mail servers** 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 UNIX mail servers** component or used external program.

Address is of the following format: **TYPE:ADDRESS**. There are three available **YPES**:

- **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).
- **pool options** — settings of a thread pool. The parameter value has a special format, described in [Special parameters types](#).
- **Lookup** — parameter value expressed as strings. In these strings, search objects (separated by commas) are specified.
- **LookupLite** — light **Lookup** (can contain only an explicit value or **Lookup** with the **file** prefix).
- **Storage** — interface for access to data store objects. It differs from the **Lookup** parameter in the list of allowed prefixes and in the fact that the `$_s` macro cannot be used.



For details on the `Lookup`, `LookupLite`, and `Storage` types, see [Lookup](#).

- `TLSSettings` — settings of the secure connection which uses TLS or SSL protocol. `TLSSettings` have a special format, described in [Special parameters types](#).
- `strings list` — parameter value expressed as a list of strings separated by commas.
If the parameter value corresponds to the `file:/path_to_file` format, the text values are taken from the specified file (in the `path_to_file` part). In this file, each text value must be specified in a separate line. If an error occurs when reading the file, information on the error is logged and program loading continues
- `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 UNIX mail servers** module outputs an error message and terminates.
- If any unknown parameter is found when loading a configuration file, **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** components to apply specified actions to objects that are detected to be malicious, suspicious or potentially dangerous.

You can specify one main action and one to three additional actions for each parameter when configuring **Dr.Web MailD** and its plug-ins. The main action is the first in the list. When configuring **Dr.Web Scanner**, only one action can be specified. Different parameters can have different available actions, they are listed in each parameter description.

You can use the following actions when configuring the settings:

- `Cure` – try to cure the infected object;
- `Remove` – delete the infected object;
- `Discard` – reject the email message without notifying the sender and delete the message;
- `Continue` – ignore the problem and continue email message processing;
- `Pass` – pass the email message to its recipient without further processing;
- `Reject` – reject the email message, delete it and notify the sender;
- `Tempfail` – notify the sender that the email message cannot temporarily be delivered and delete the message;



The following additional actions are available:

- Quarantine – move the email message to the **Quarantine** folder;
- Redirect [`(address[|address|...])`] – redirect the email message to the address specified within the brackets. If no address is specified, the message is redirected according to the **RedirectMail** parameter value in the [MailD] [section](#) of **Dr.Web MailD** configuration file. You can specify several addresses, separating them by the "|" character;
- Notify – send a report about detected threats, message processing is not stopped;
- Score (score) – add a SCORE to the message counter. The SCORE value can be negative;
- Add-header (HEADER) – add a header of the following type `[NAME:]BODY` to the email message, where NAME – is the name of the header (the default name is X-DrWeb-MailD) and BODY is the text of the header.

Please note that you can use strings from language files (`.lng`). String to be inserted is defined by a number, for example:

```
add-header (X-Added-Header:$3)
```

In this case, the X-Added-Header header is to be added with the `<value>` value which is taken from the `3="<value>"` string of the used [language file](#).

If you use ";", "(" and ")" characters, it is necessary to escape them in order to avoid incorrect interpretation of the header.

To escape characters:

To escape a punctuation character in a header, use 3 backslashes "\".

Example:

```
EmptyFrom = continue, add-header (header:Empty header\\\; spam)
```

To escape parenthesis, use a backslash "\".

Example:

```
ProcessingErrors = tempfail,add-header (\(header:header\))
```

To escape a whole header, use quotes: "add-header (BODY)".

Example:

```
ProcessingErrors = tempfail,"add-header (header:(spam))"
```

To escape double quotation marks, use 3 backslashes "\".

Examples:

```
EmptyFrom = continue,"add-header (header[X-Header]:new\\\\"header\\\\")"
EmptyFrom = continue,add-header (header\[X-Header\]:new\\\\"header\\\\")
```

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.



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 UNIX mail servers** 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.

If previously the product was installed from packages of other formats (for example, RPM or DEB), ensure that they are carefully uninstalled.

Dr.Web for UNIX mail servers 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 UNIX mail servers** components.

You can install, deinstall, and update **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers solution is distributed as a self-extracting package `drweb-mail-[product name]_[version number]~[OS name].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`: contains 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-maild`: contains **Dr.Web MailD** executable files and its documentation. It requires `drweb-maild-common` package to be installed;
- `drweb-maild-common`: contains libraries for **Dr.Web Agent**, **Dr.Web Monitor** and **Dr.Web MailD**. It requires `drweb-common`, `drweb-gperftools0`, `drweb-agent` and `drweb-monitor` packages to be installed;
- `drweb-maild-plugin-drweb`: contains library of **Drweb** plug-in, its configuration file, documentation and configuration script. It requires `drweb-maild` package to be installed;
- `drweb-maild-web`: contains web interface of **Dr.Web for UNIX mail servers**;
- `drweb-maild-plugin-headersfilter`: contains library of **Headersfilter** plug-in, its configuration file, documentation and configuration script. It requires `drweb-maild` package to be installed;
- `drweb-maild-plugin-modifier`: contains library of **Modifier** plug-in, its configuration file, documentation and configuration script. It requires `drweb-maild` package to be installed;
- `drweb-maild-plugin-vaderetro`: contains configuration file of **Vaderetro** plug-in, documentation and configuration script. It requires `drweb-maild` and `drweb-libvaderetro` packages to be installed;
- `drweb-libvaderetro`: contains library of **Vaderetro** plug-in;
- `drweb-maild-smtp`: contains executable files of **Sender** and **Receiver** modules which enable operation of **Dr.Web for UNIX mail servers** as a proxy-server for SMTP and LMTP protocols, **Dr.Web MailD** configuration file with corresponding settings, documentation and configuration script for **Dr.Web Monitor**. It requires `drweb-maild` package to be installed;
- `drweb-maild-cgp`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Communigate Pro** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **Communigate Pro** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-maild-courier`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Courier** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **Courier** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-maild-exim`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Exim** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **Exim** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-maild-postfix`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Postfix** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **Postfix** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-maild-qmail`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Qmail** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **Qmail** for interaction with **Dr.Web**



MailD. It requires `drweb-maild` package to be installed;

- `drweb-maild-sendmail`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **Sendmail** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script for adjustment of **Sendmail** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-maild-zmailer`: contains executable files of **Sender** and **Receiver** modules which enable interaction with **ZMailer** mail transfer agent, **Dr.Web MailD** configuration file with settings for the specific MTA, documentation and configuration script of **ZMailer** for interaction with **Dr.Web MailD**. It requires `drweb-maild` package to be installed;
- `drweb-gperftools0`: contains **Google Performance Tools** library used by **Dr.Web MailD**. It requires `drweb-libs` package to be installed;
- `drweb-mail-servers-gateways-doc`: contains **Administrator manual** in English and Russian languages.

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 UNIX mail servers** 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-mail-[product name]_[version number]~[OS name].run
```

and then run it:

```
# ./drweb-mail-[product name]_[version number]~[OS name].run
```

As a result,

`drweb-mail-[product name]_[version number]~[OS name]`

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-mail-[product name]_[version number]~[OS name].run --noexec
```

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

```
# drweb-mail-[product name]_[version number]~[OS name]/install.sh
```

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

```
# drweb-mail-[product name]_[version number]~[OS name]/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.



- If you select the **Run interactive postinstall script** check box in the corresponding window of the GUI installer, then after installation of the components completes, the post-install script is initialized for **Dr.Web for UNIX mail servers** basic adjustment.



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-mail-[product name]_[version number]~[OS name]` directory and `.run` file after successful completion of installation.

Using GUI Installer

To install with GUI

1. Enter the following command:

```
# drweb-mail-[product name]_[version number]~[OS name]/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**.

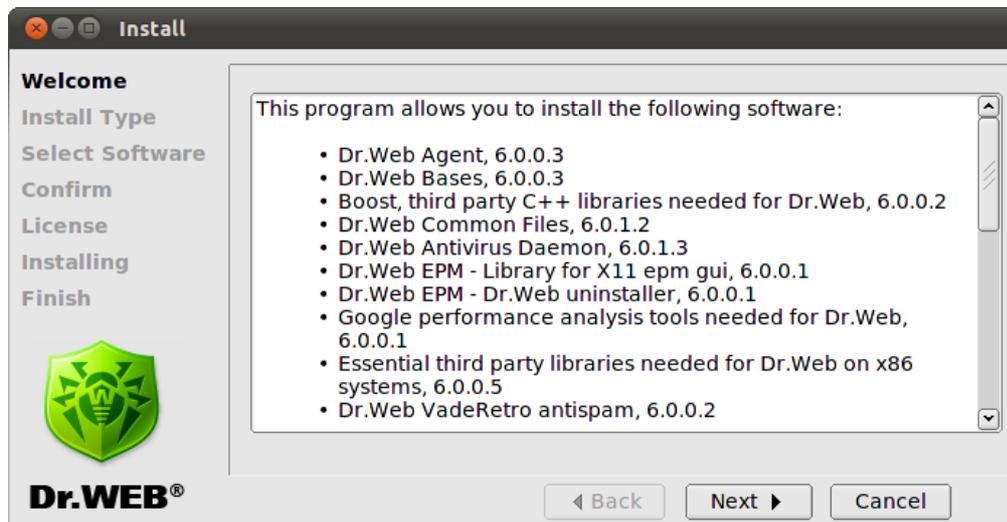


Figure 2. Welcome screen

2. On the **Install Type** screen, select the installation type: typical configuration for **Dr.Web for Mail Gateways** or certain MTA **Dr.Web for <MTA> (Full installation)** with all the necessary components selected by default or custom configuration.

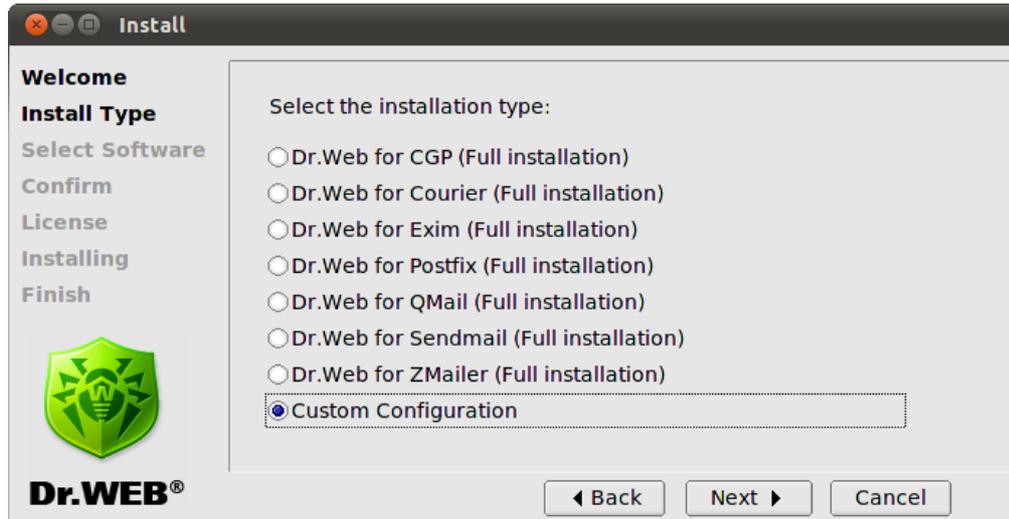


Figure 3. Install Type screen for MTA

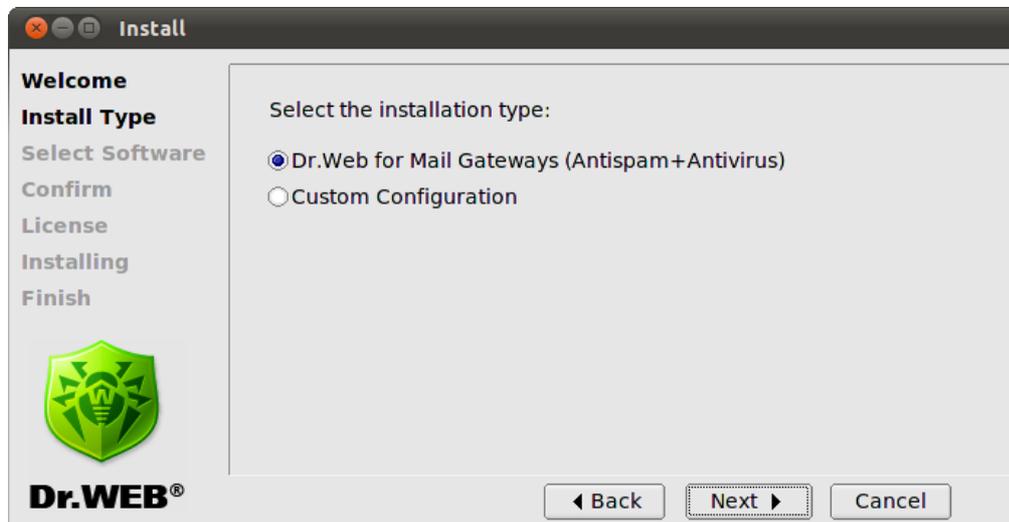


Figure 4. Install Type screen for **Dr.Web for Mail Gateways**

If you selected **Custom Configuration**, then select necessary components on the **Select Software** screen:

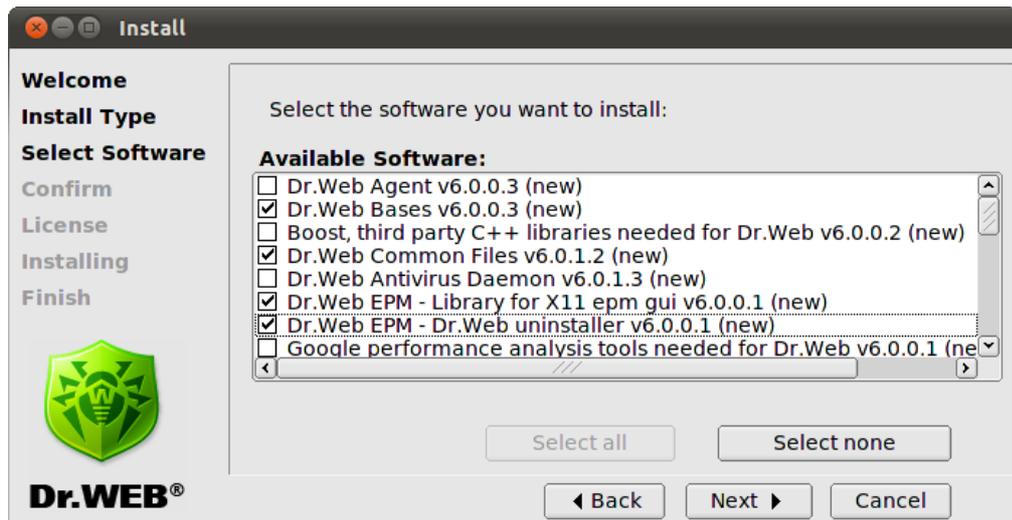


Figure 5. Select Software screen



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.

During installation, packages for different mail transfer agents can conflict with each other (`drweb-maild-smtp` and various `drweb-maild-<MTA>`). For example, if you try to install **Dr.Web Mail Daemon – Exim Connector** and **Dr.Web Mail Daemon – Postfix Connector** simultaneously, you will receive an error message and suggestion to select only one of them.

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:

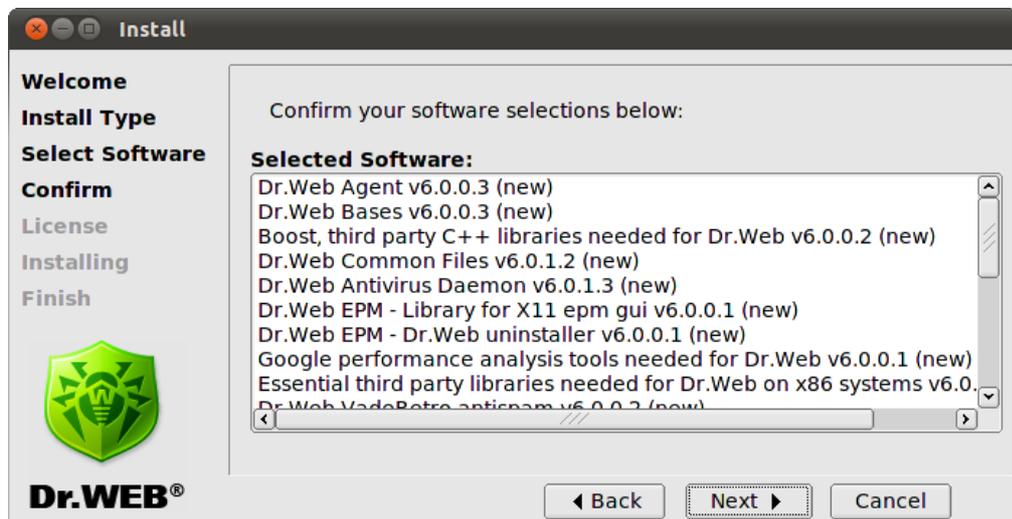


Figure 6. Confirm screen

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):

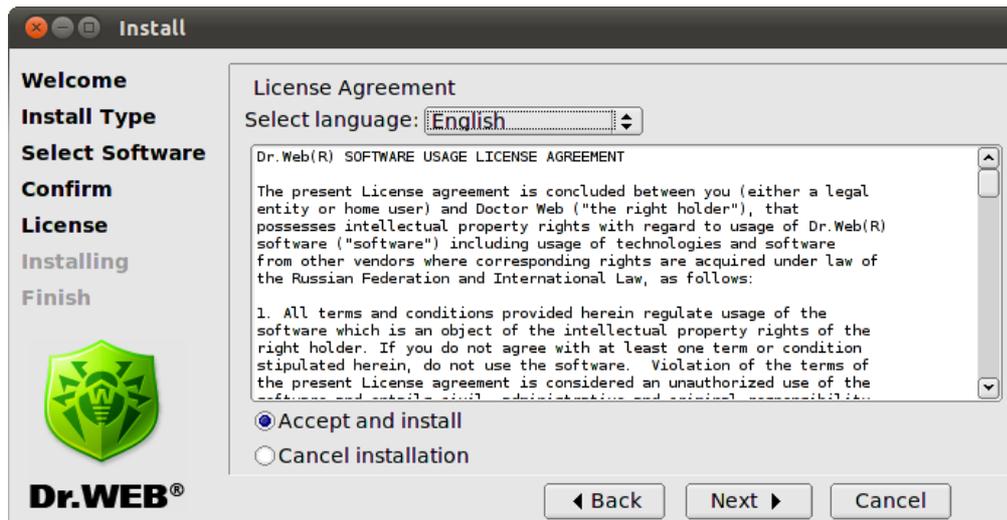


Figure 7. License Agreement screen

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

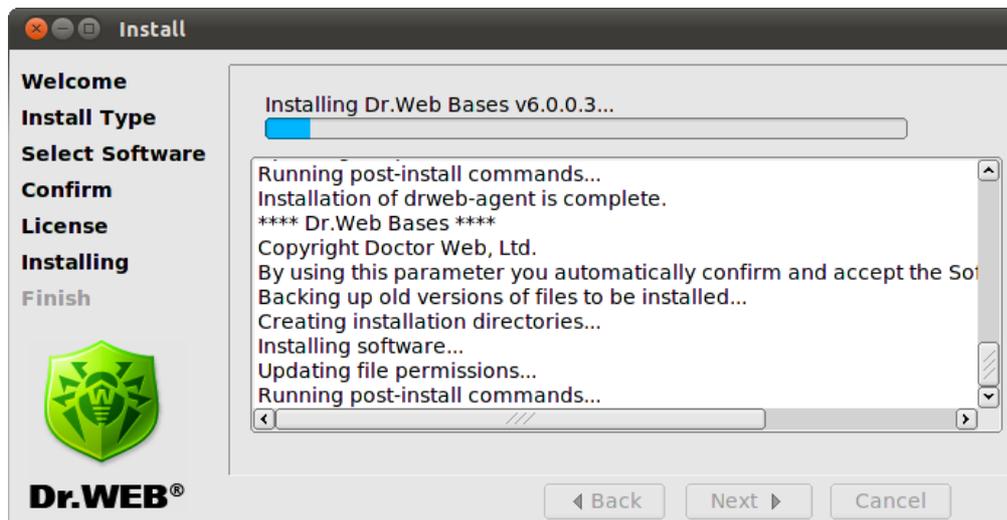


Figure 8. Installing screen

This report is logged at the same time in the `install.log` log file located at the `drweb-mail-[product name]_[version number]~[OS name]` directory. If you selected **Run interactive post-install script**, once component installation completes, the post-install script for **Dr.Web for UNIX mail servers** basic configuration initializes.



```
DrWeb
This installation script will help you to configure DrWeb for Mail server Antivirus+Antispam
Do you want to continue? (YES/no)
yes
Enter list of plugins to process message before placing it to queue/DB.
Possible values: (headersfilter|modifier). Values are delimited with commas.
[default=]:modifier
Enter list of plugins to process message after placing it to queue/DB.
Possible values: (headersfilter). Values are delimited with commas.
[default=]:headersfilter
Enter email address to send notifications to.
[default=postmaster@localhost]:
Enter email address to send notifications from.
[default=DrWEB-MAIL-DAEMON@localhost]:
Enter list of protected networks (e.g. 127.0.0.0/8). Values are delimited with commas.
[default=127.0.0.0/8]:
Enter list of protected domains. Values are delimited with commas.
[default=localhost]:
Enter language(s) to use in reports.
Possible values: (en|ja|ru). Values are delimited with commas.
[default=en]:

=====
Configuration:
Plugins directory = /opt/drweb/maild/plugins
Ing files directory = /etc/drweb/maild/Ing
Before queue plugins = modifier
After queue plugins = headersfilter
Administrator email address = postmaster@localhost
Filter email address = DrWEB-MAIL-DAEMON@localhost
Protected networks = 127.0.0.0/8
Protected domains = localhost
Language(s) for reports = en

Press 1 to Save updated configuration,
2 to go Back to editing, 3 to Cancel or 5 to Redisplay [1]
█
```

Figure 9. Interactive post-install script

After initialization of the script, you can specify a path to the key file, set an order of mail processing by the plug-ins and automatically enable services necessary for **Dr.Web for UNIX mail servers** proper operation (for example, **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**). You can also specify the list of protected networks and domains.



```
DrWeb
Protected domains = localhost
Language(s) for reports = en

Press 1 to Save updated configuration,
2 to go Back to editing, 3 to Cancel or 5 to Redisplay [1]
1
General/Hostname = localhost
Notifier/AdminMail = postmaster@localhost
Mail/RedirectMail = postmaster@localhost
Notifier/FilterMail = DrWEB-MAIL-DEMON@localhost
Filters/AfterQueueFilters = headersfilter
Filters/BeforeQueueFilters = modifier
Mail/ProtectedNetworks = 127.0.0.0/8
Mail/ProtectedDomains = localhost
Notifier/NotifyLangs = en
Monitor/RunAppList = MAILD

/etc/drweb/monitor.conf patched OK.
/etc/drweb/mailed_postfix.conf patched OK.

Do you want to configure MTA for DrWeb for Mail server Antivirus+Antispam? (YES/no)
yes

-----
Welcome to the Dr.Web InstallShield Wizard.

The InstallShield Wizard will configure POSTFIX.

Perform MTA configuration?
Please enter yes or no.
yes

Error: the Postfix configuration file /etc/postfix/master.cf was not found!
Info: you can specify the MTA_CONFIG_PATH environment variable.
Please, refer to documentation on POSTFIX adjustment residing in /opt/drweb/doc/mailed directory.

Do you want to configure services? (YES/no)
yes
Configuring startup of drwebd...
Already running.
Configuring startup of drweb-monitor...
Already running.

Configuration completed successfully.
Press Enter to finish.
```

Figure 10. Configuring MTA and starting services automatically

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

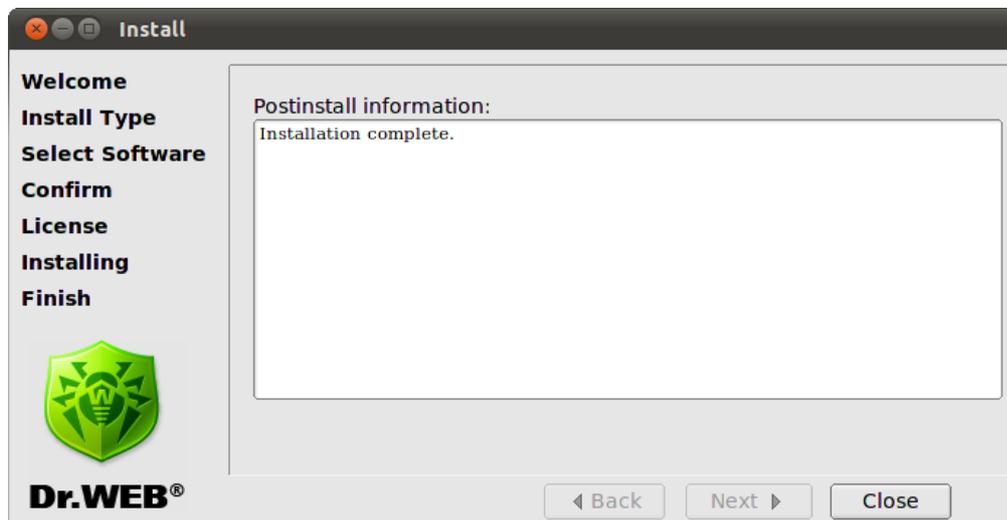


Figure 11. Finish screen

Using Console Installer

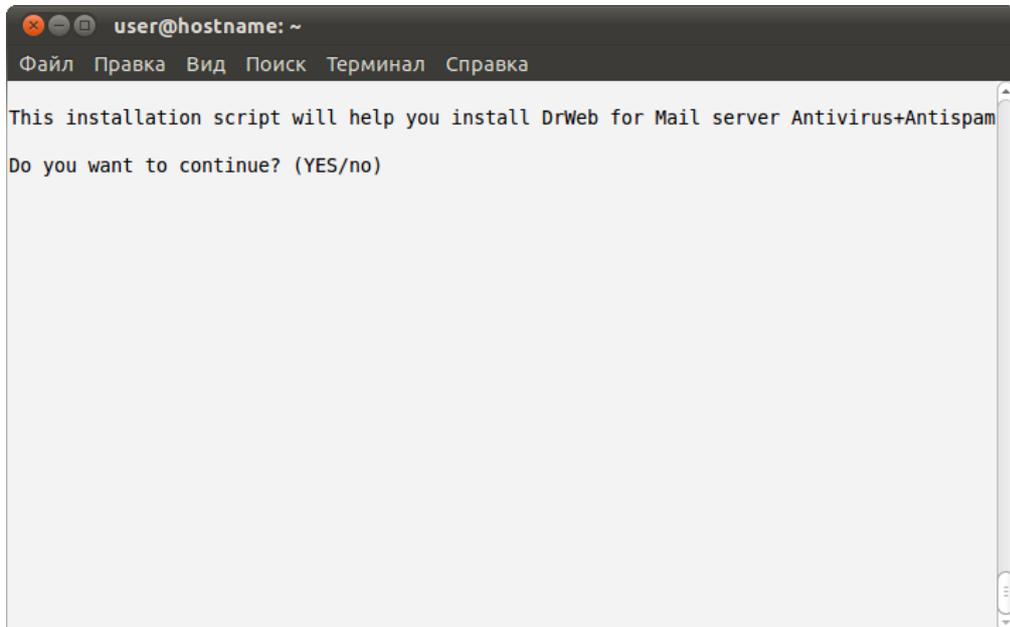
Console installer starts automatically if the GUI installer fails to start. If the console installer also fails to start (for example, if it is impossible to gain necessary privileges), you can try to run the following command with `root` privileges:



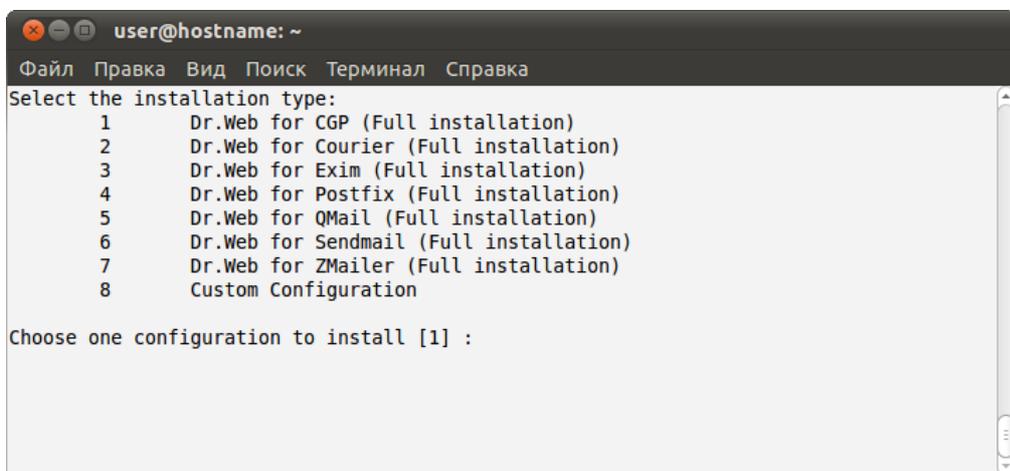
```
# drweb-mail-[product name]_[version number]~[OS name]/setup.sh
```

To install from console

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



2. If you want to install **Dr.Web for UNIX mail servers**, enter **Y** or **Yes** (values are case insensitive), otherwise enter **N** or **No**. Press ENTER.
3. If you chose to install **Dr.Web for UNIX mail servers**, installer suggests you to select the installation type:



To select a required mode, enter the respective number and press ENTER.

4. If you selected **Custom Configuration**, specify required components to install:



```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
[ ] 16 Dr.Web Mail Daemon - Dr.Web plugin v6.0.0.2 (new)
[ ] 17 Dr.Web Mail Daemon - HeadersFilter plugin v6.0.0.2 (new)
[ ] 18 Dr.Web Mail Daemon - Modifier plugin v6.0.0.2 (new)
[ ] 19 Dr.Web Mail Daemon - VadeRetro plugin v6.0.0.2 (new)
[ ] 20 Dr.Web Mail Daemon - Postfix connector v6.0.0.2 (new)
[ ] 21 Dr.Web Mail Daemon - qmail connector v6.0.0.2 (new)
[ ] 22 Dr.Web Mail Daemon - Sendmail connector v6.0.0.2 (new)
[ ] 23 Dr.Web Maild Web Interface v6.0.0.2 (new)
[ ] 24 Dr.Web Mail Daemon - ZMailer connector v6.0.0.2 (new)
[ ] 25 Dr.Web Mail Daemon v6.0.0.2 (new)
[ ] 26 Dr.Web Monitor v6.0.0.3 (new)
[ ] 27 Dr.Web Antivirus Scanner v6.0.1.3 (new)
[ ] 28 Dr.Web Updater v6.0.0.4 (new)

To select a package you want to install or deselect some previously
selected package - enter the corresponding package number and press Enter.

You may enter A or All to select all the packages, and N or None to deselect all of the
m.
Enter I or Install to install selected packages.
Enter 0, Q or Quit to quit the dialog.
All values are case insensitive.
Select:
```

To specify a required component, enter the respective number and press ENTER.

5. Review the **License Agreement**. To scroll the text, press SPACEBAR:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Dr.Web(R) SOFTWARE USAGE LICENSE AGREEMENT

The present License agreement is concluded between you (either a legal
entity or home user) and Doctor Web ("the right holder"), that
possesses intellectual property rights with regard to usage of Dr.Web(R)
software ("software") including usage of technologies and software
from other vendors where corresponding rights are acquired under law of
the Russian Federation and International Law, as follows:

1. All terms and conditions provided herein regulate usage of the
software which is an object of the intellectual property rights of the
right holder. If you do not agree with at least one term or condition
stipulated herein, do not use the software. Violation of the terms of
the present License agreement is considered an unauthorized use of the
software and entails civil, administrative and criminal responsibility.

2. If you are a legal owner of the Software's copy, you receive the
--More-- (24%)
```

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.

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



```
user@hostname: ~
Файл Правка Вид Поиск Терминал Справка
Creating installation directories...
Installing software...
Updating file permissions...
Running post-install commands...
Installation of drweb-libs is complete.
Backing up old versions of files to be installed...
Creating installation directories...
Installing software...
Updating file permissions...
Installation of drweb-boost144 is complete.
Backing up old versions of files to be installed...
Creating installation directories...
Installing software...
Checking configuration files...
Updating file permissions...
Running post-install commands...
Installation of drweb-agent is complete.
Copyright Doctor Web, Ltd.
```

7. Once installation of the components completes, the post-install script runs automatically to set up **Dr.Web for UNIX mail servers** 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 UNIX mail servers** proper operation (for example, **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**). In addition, you can set an order of mail processing by the plug-ins and specify lists of protected networks and domains.

```
user@hostname: ~
Файл Правка Вид Поиск Терминал Справка

This installation script will help you to configure DrWeb for Mail server Antivirus+Antispam

Do you want to continue? (YES/no) yes
yes

Enter path to key file for Dr.Web MailD.
If you don't have the key yet you can leave this value unspecified,
but you must set LicenseFile parameter value in configuration file agent.conf,
and parameter Key in configuration file drweb32.ini before MailD is
launched or any plugin is installed.
[default=]:

Enter list of plugins to process message before placing it to queue/DB.
Possible values: (vaderetro|headersfilter|drweb|modifier). Values are delimited with commas.
[default=headersfilter]:headersfilter

Enter list of plugins to process message after placing it to queue/DB.
Possible values: (vaderetro|drweb|modifier). Values are delimited with commas.
[default=modifier]:
```

Removing Distribution Package for UNIX Systems

To remove all the components of **Dr.Web for UNIX mail servers** 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**.

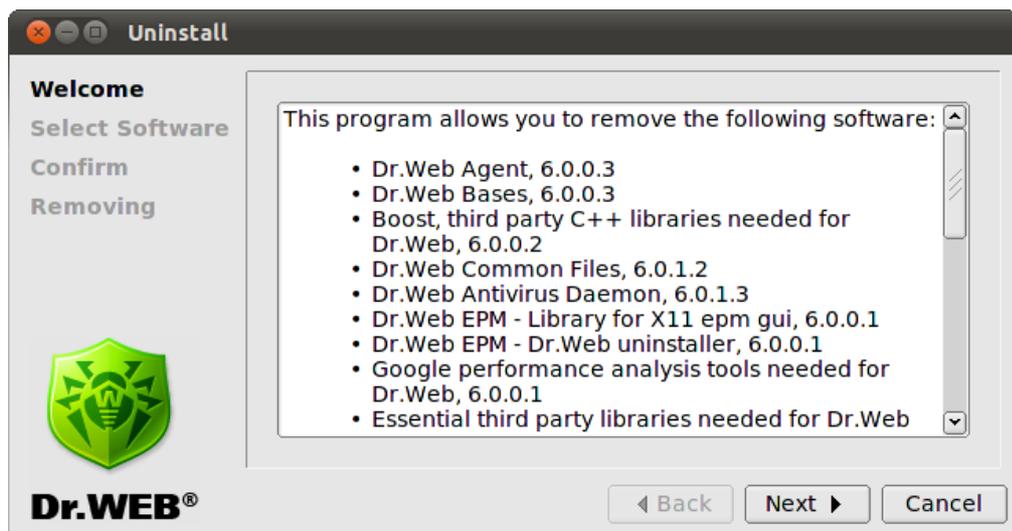


Figure 12. Welcome screen

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

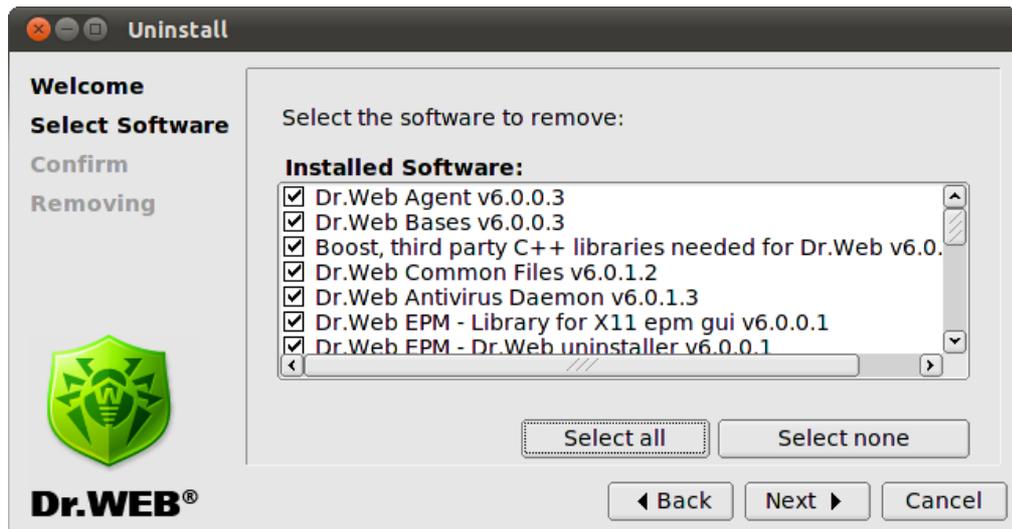


Figure 13. Select Software screen

All corresponding dependencies are selected to be uninstalled automatically.

If you installed **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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:

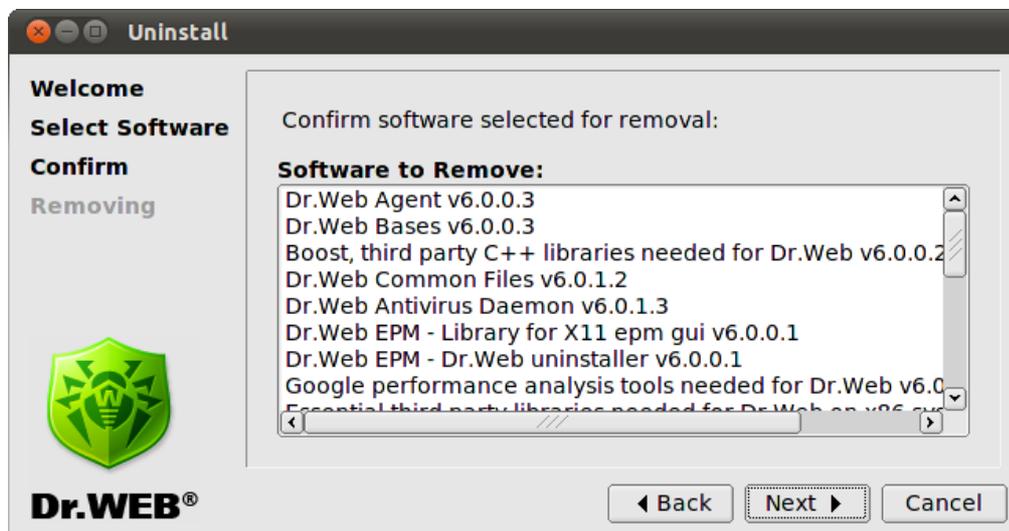


Figure 14. Confirm screen

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:

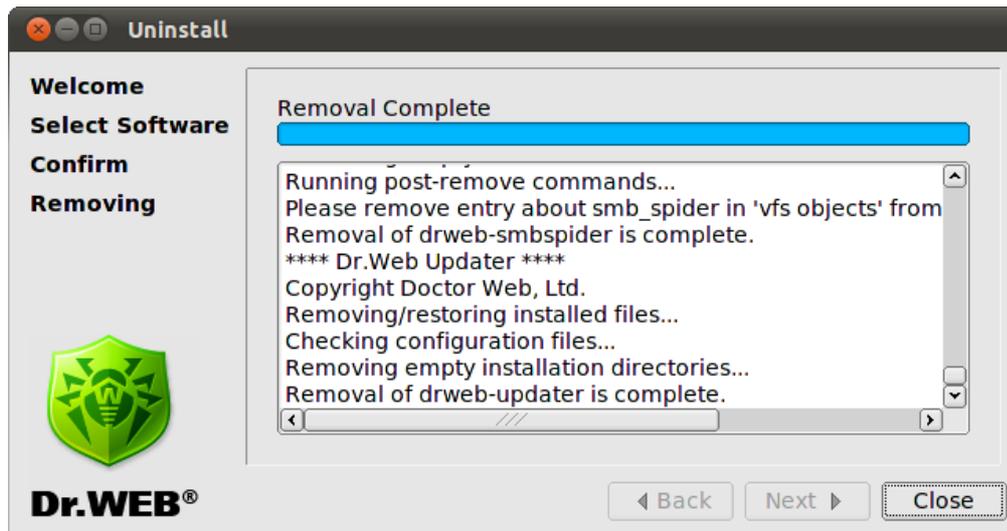


Figure 15. Removing screen

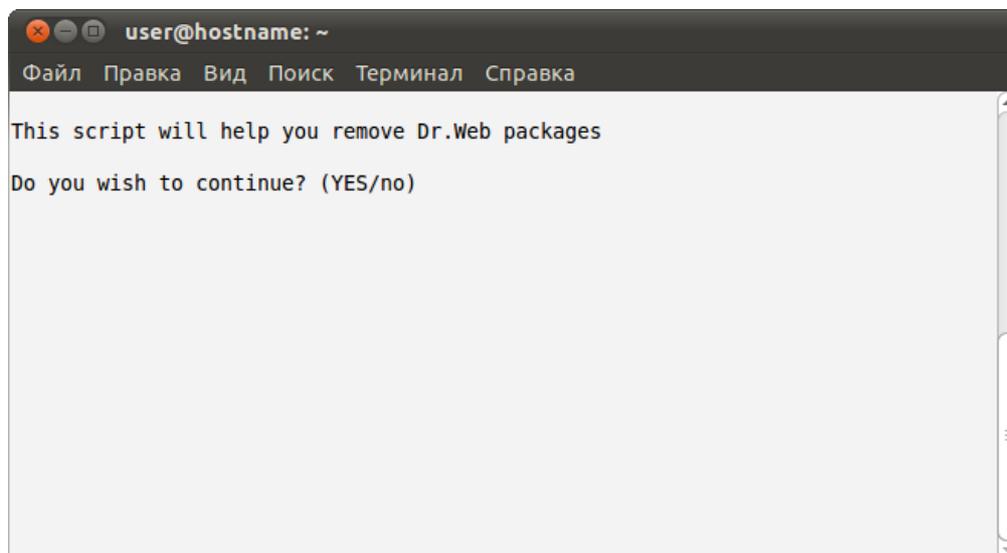
5. Click **Close** to exit setup.

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 UNIX mail servers**, enter **yes**, otherwise enter **no**. Press ENTER.

2. Review the list of components available for removal:



```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
[X] 10 Dr.Web VadeRetro antispam (6.0.0.2)
[X] 11 Dr.Web Mail Daemon - CommuniGate Pro connector (6.0.0.2)
[X] 12 Dr.Web Mail Daemon - common files (6.0.0.2)
[X] 13 Dr.Web Mail Daemon - Dr.Web plugin (6.0.0.2)
[X] 14 Dr.Web Mail Daemon - HeadersFilter plugin (6.0.0.2)
[X] 15 Dr.Web Mail Daemon - Modifier plugin (6.0.0.2)
[X] 16 Dr.Web Mail Daemon - VadeRetro plugin (6.0.0.2)
[X] 17 Dr.Web Mail Daemon (6.0.0.2)
[X] 18 Dr.Web Maild Web Interface (6.0.0.2)
[X] 19 Dr.Web mail server and mail gateways documentation (6.0.0.2)
[X] 20 Dr.Web Monitor (6.0.0.3)
[X] 21 Dr.Web Antivirus Scanner (6.0.1.3)
[X] 22 Dr.Web Updater (6.0.0.4)

To select a package you want to remove or deselect some previously
selected package - enter the corresponding package number and press Enter.

You may enter A or All to select all the packages, and N or None to deselect all of the
m.
Enter R or Remove to remove selected packages.
Enter 0, Q or Quit to quit the dialog.
All values are case insensitive.
Select:
```

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:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
drweb-agent
drweb-bases
drweb-boost144
drweb-common
drweb-daemon
drweb-epm6.0.0-libs
drweb-epm6.0.0-uninst
drweb-gperftools0
drweb-libs
drweb-libvaderetro
drweb-maild-cgp
drweb-maild-common
drweb-maild-plugin-drweb
drweb-maild-plugin-headersfilter
drweb-maild-plugin-modifier
drweb-maild-plugin-vaderetro
drweb-maild
drweb-monitor
drweb-scanner
drweb-updater
Are you sure you want to remove the selected packages? (YES/no)
```

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

Installing from Native Packages

You can install **Dr.Web for UNIX mail servers** from native packages for common **Linux** distributions or **FreeBSD** operating system.

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.

Depending on required solution, replace `<package name>` in commands below with one of the following packages:

- `drweb-mail-gateways-as` – **Dr.Web Anti-spam for UNIX Mail Gateways;**
- `drweb-mail-gateways-av` – **Dr.Web Anti-virus for UNIX Mail Gateways;**
- `drweb-mail-gateways-av-as` – **Dr.Web Anti-virus and Anti-spam for UNIX Mail Gateways;**
- `drweb-courier-as` – **Dr.Web Anti-spam for Courier Mail Servers;**
- `drweb-courier-av` – **Dr.Web Anti-virus for Courier Mail Servers;**
- `drweb-courier-av-as` – **Dr.Web Anti-virus and Anti-spam for Courier Mail Servers;**
- `drweb-postfix-as` – **Dr.Web Anti-spam for Postfix Mail Servers;**
- `drweb-postfix-av` – **Dr.Web Anti-virus for Postfix Mail Servers;**
- `drweb-postfix-av-as` – **Dr.Web Anti-virus and Anti-spam for Postfix Mail Servers;**
- `drweb-qmail-as` – **Dr.Web Anti-spam for Qmail Mail Servers;**
- `drweb-qmail-av` – **Dr.Web Anti-virus for Qmail Mail Servers;**
- `drweb-qmail-av-as` – **Dr.Web Anti-virus and Anti-spam for Qmail Mail Servers;**
- `drweb-sendmail-as` – **Dr.Web Anti-spam for Sendmail Mail Servers;**
- `drweb-sendmail-av` – **Dr.Web Anti-virus for Sendmail Mail Servers;**
- `drweb-sendmail-av-as` – **Dr.Web Anti-virus and Anti-spam for Sendmail Mail Servers;**
- `drweb-cgp-as` – **Dr.Web Anti-spam for CommuniGate Pro Mail Servers;**
- `drweb-cgp-av` – **Dr.Web Anti-virus for CommuniGate Pro Mail Servers;**
- `drweb-cgp-av-as` – **Dr.Web Anti-virus and Anti-spam for CommuniGate Pro Mail Servers;**
- `drweb-exim-as` – **Dr.Web Anti-spam for Exim Mail Servers;**
- `drweb-exim-av` – **Dr.Web Anti-virus for Exim Mail Servers;**
- `drweb-exim-av-as` – **Dr.Web Anti-virus and Anti-spam for Exim Mail Servers;**
- `drweb-zmailer-as` – **Dr.Web Anti-spam for ZMailer Mail Servers;**
- `drweb-zmailer-av` – **Dr.Web Anti-virus for ZMailer Mail Servers;**
- `drweb-zmailer-av-as` – **Dr.Web Anti-virus and Anti-spam for ZMailer Mail Servers.**



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.

Debian, Ubuntu (apt)

1. Installation:

Debian repository is signed with the digital key. It is necessary to import the key or correct operation. To do this, use the following command

```
wget -O - http://officeshield.drweb.com/drweb/drweb.key | apt-key add -
```



or

```
curl http://officeshield.drweb.com/drweb/drweb.key | apt-key add -
```

To add the repository to your system, add the following line to `/etc/apt/sources.list` file:

```
deb http://officeshield.drweb.com/drweb/debian stable non-free
```

To install **Dr.Web for UNIX mail servers**, use the following commands:

```
apt-get update
apt-get install <package name>
```

2. Deinstallation:

To remove **Dr.Web for UNIX mail servers**, use the following command:

```
apt-get remove <package name>
```

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: '*'):

```
apt-get remove drweb*
```

To automatically remove unused packages from the system, use the following command:

```
apt-get autoremove
```



Removal with the use of **apt-get** has the following features:

1. The first variant of the command removes only the `<package name>` 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 which names start with 'drweb' (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 UNIX mail servers**.
3. The third variant of the command removes from the system all unused packages which were automatically installed for resolving dependences of some removed packages. Please note that this command removes all unused packages from the system, not only those of **Dr.Web for UNIX mail servers**.

You can also use alternative package managers (for example, **Synaptic**, **aptitude**) to install or remove the packages. Moreover, it is recommended to use alternative managers, such as **aptitude**, to resolve a package conflict if it occurs.

ALT Linux, PCLinuxOS (apt-rpm)

1. Installation:

To add the repository to you system, add the following line to the `/etc/apt/sources.list` file:

32-bit version:

```
rpm http://officeshield.drweb.com/drweb/altlinux stable/i386 drweb
```

64-bit version:

```
rpm http://officeshield.drweb.com/drweb/altlinux stable/x86_64 drweb
```



To install **Dr.Web for UNIX mail servers**:

```
apt-get update
apt-get install <package name>
```

2. Uninstallation:

In this case, uninstallation process is the same as for **Debian** and **Ubuntu** (see above).

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

Mandriva (urpmi)

1. Installation:

Download a repository key from <http://officeshield.drweb.com/drweb/drweb.key> and save it to the disk. After that, import the key with the following command:

```
rpm --import <path to repository key>
```

Open the following file:

<http://officeshield.drweb.com/drweb/drweb-i386.urpmi-media>

or

http://officeshield.drweb.com/drweb/drweb-x86_64.urpmi-media

After you open a file, you will be offered to add a repository to the system.

Alternatively, you can add the repository via console using one of the following commands:

```
urpmi.addmedia drweb http://officeshield.drweb.com/drweb/mandriva/stable/i386/
```

or

```
urpmi.addmedia drweb http://officeshield.drweb.com/drweb/mandriva/stable/x86_64/
```

To install **Dr.Web for UNIX mail servers**:

```
urpmi.update drweb
urpmi <package name>
```

2. Deinstallation:

To remove **Dr.Web for UNIX mail servers**:

```
urpme <package name>
```

To automatically remove unused packages from the system:

```
urpme --auto-orphans <package name>
```



Removal with the use of **urpme** has the following features:

1. The first variant the command removes only the `<package name>` 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 the `<package name>` package and all unused packages, which were automatically installed to resolve dependences of some removed packages. Please note that this command removes all unused packages from the system, not only those of **Dr.Web for UNIX mail servers**.



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

Red Hat Enterprise Linux, Fedora, CentOS (yum)

1. Installation:

Add to the `/etc/yum.repos.d` directory the file with following content:

32-bit version:

```
[drweb]
name=DrWeb - stable
baseurl=http://officeshield.drweb.com/drweb/el5/stable/i386/
gpgcheck=1
enabled=1
gpgkey=http://officeshield.drweb.com/drweb/drweb.key
```

64-bit version:

```
[drweb]
name=DrWeb - stable
baseurl=http://officeshield.drweb.com/drweb/el5/stable/x86_64/
gpgcheck=1
enabled=1
gpgkey=http://officeshield.drweb.com/drweb/drweb.key
```

To install **Dr.Web for UNIX mail servers**:

```
yum install <package name>
```

2. Deinstallation:

To remove **Dr.Web for UNIX mail servers**:

```
yum remove <package name>
```

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: '*'):

```
yum remove drweb*
```



Removal with the use of **yum** has the following features:

1. The first variant of the command removes only the `<package name>` 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 UNIX mail servers**.

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

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/e15/stable/x86_64/ drweb
```

To install **Dr.Web for UNIX mail servers**, use the following commands:

```
zypper refresh
zypper install <package name>
```

2. Deinstallation:

To remove **Dr.Web for UNIX mail servers**, use the following command:

```
zypper remove <package name>
```

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 <package name>, 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 UNIX mail servers**.

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

FreeBSD operating system

Installation:

You can install **Dr.Web** products from meta-ports for **FreeBSD**. Download the drweb-maild-meta_current-current~freebsd_all.tar.gz archive from <http://officeshield.drweb.com/drweb/freebsd/ports/>. After that, unpack the archive and use the make install command to compile and install **Dr.Web for UNIX mail servers**. If you install **Dr.Web for UNIX mail servers** in **FreeBSD** 6.1, specify the path to the /usr/ports/Mk/ directory using the -I parameter. That directory contains the ports tree.

Example:

```
tar -xzvf drweb-maild-meta_current-current~freebsd_all.tar.gz
make install -I /usr/ports/Mk/
```



Please note that after you update **Dr.Web for UNIX mail servers** from native packages, it is required to restart the suite. To do that, reload **Dr.Web Monitor** by running the following command: `/etc/init.d/drweb-monitor restart`.

An attempt to reload the module by sending a SIGHUP signal will cause an error if the plug-in libraries were updated.



Configuration Scripts

After installation of the components, you can use the `configure.pl` configuration script to setup basic configuration of **Dr.Web MailD**. This script is located in the `%bin_dir/maild/scripts/` directory.

On startup, the script offers you to specify:

- order of mail processing with a certain plug-in (for example, whether it should receive messages before or after they are added to the database);
- language to use for notifications and an address where they are to be sent;
- paths to the lists of protected networks and domains.

Also you should use the `configure_mta.sh` script. This script is responsible for setting up interaction between **Dr.Web for UNIX mail servers** and the currently used mail system. After startup, the script checks whether the required mail system is installed. If it appears to be missing, the script finishes its operation. If the required mail system is installed, the script asks the user several questions on essential settings for basic setup and makes the necessary changes in corresponding configuration files.

This information is enough to start **Dr.Web for UNIX mail servers**, but for a full-featured performance it is required to configure each component and MTA manually.

For detailed information on parameters, methods, and techniques, see the corresponding chapters of this Manual: [Adjustment and Startup](#), [Plug-ins](#), [Integration with Mail Transfer Systems](#).



The `configure_mta.sh` and `plugin_NAME_configure.pl` scripts from the `%bin_dir/maild/scripts/` directory do not provide full-featured configuration for best performance of plug-ins and MTA. Use these scripts as reference source only.



Starting Dr.Web for UNIX mail servers

This section describes startup of **Dr.Web for UNIX mail servers** in **Linux**, **Solaris** or **FreeBSD** operating systems.

For Linux and Solaris OS

To run **Dr.Web for UNIX mail servers**:

1. Register the software.
2. Copy or move the key file to the directory with **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail**



servers components (**Sender**, **Receiver**, **Notifier** etc.).

In case of installation from native packages in Solaris:

During **Dr.Web for UNIX mail servers** 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 UNIX mail servers**), 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 UNIX mail servers** from native packages in Solaris, run **Dr.Web for UNIX mail servers** 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 UNIX mail servers**:

1. Register the software.
2. Copy or move the key file (with the `.key` extension) to the directory with **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** components (**Sender**, **Receiver**, **Notifier** etc.).

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 UNIX mail servers**, 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 UNIX mail servers** 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`:

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

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

- o 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 UNIX mail servers**.

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 UNIX mail servers** are specified in the key file.

License key file contains the following information:

- list of **Dr.Web for UNIX mail servers** components licensed to the user;
- license period;
- list of plug-ins licensed to the user (some plug-ins do not require registration in the key file);
- other restrictions (for example, number of emails to be checked by plug-ins per day).

By default, the license key file is located in the directory with **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers**. 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers provides you with the possibility to use several license key files simultaneously. List of plug-ins licensed to the user includes all plug-ins mentioned in the key files (or at least in one of them). Limitations on operation of a certain plug-in are composed of restrictions specified in all used key files.

During operation of the suite, various plug-ins must have the same limitations. In case there are several key files with different limitations, operation of the plug-ins is restricted to the minimal permission specified in the used key files.

Example:

Let us assume that three license key files are used. In the first key file, limitation for **Drweb** plug-in is set to 10,000 letters per day. In the second key file, limitation for **Vaderetro** plug-in is set to 15,000 letters per day. In the third key file, limitation for **Drweb** plug-in is set to 10,000 letters per day. Thus, **Dr.Web for UNIX mail servers** will be able to work with both plug-ins mentioned in key files, but total limitation on operation of these plug-ins will be set to 15,000 letters per day (as for **Vaderetro**), in spite of the fact that **Drweb** plug-in can actually process 20,000 letters per day.



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



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 <code>/dev</code> directory and named as <code>sdx</code> or <code>hdX</code> , where <code>x</code> is a letter of the Latin alphabet (<code>a</code> , <code>b</code> , <code>c</code> , ...). For example: <code>hda</code> , <code>sda</code> . Thus, to check MBR of disk <code>sda</code> , specify the following: <code>disk:///dev/sda</code>
<code>-@[+]<file></code>	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.
<code>--</code>	Instructs to read the list of objects for scanning from the standard input stream (<code>stdin</code>).
<code>-sd</code>	Sets recursive search for files to scan in subfolders.
<code>-fl</code>	Instructs to follow symbolic links to both files and folders. Links that cause loops are ignored.
<code>-mask</code>	Instructs to ignore filename masks.

Diagnostic Parameters

These parameters determine object types to be scanned for viruses:

Parameter	Description
<code>-al</code>	Instructs to scan all objects defined by scan paths regardless of their file extension and structure. This parameter is opposite to the <code>-ex</code> parameter.
<code>-ex</code>	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 <code>-ini</code> 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 <code>-al</code> parameter.
<code>-ar[d m r][n]</code>	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 (<code>d</code> , <code>m</code> or <code>r</code>) 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.
<code>-cn[d m r][n]</code>	Instructs to scan files within containers (HTML, RTF, PowerPoint). If additional modifiers (<code>d</code> , <code>m</code> or <code>r</code>) 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.
<code>-ml[d m r][n]</code>	Instructs to scan contents of mail files. If additional modifiers (<code>d</code> , <code>m</code> or <code>r</code>) 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.
<code>-upn</code>	Scan executable files packed with LZEXE, DIET, PKLITE, EXEPACK without output of the compression type.
<code>-ha</code>	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 action applied 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
<code>-v, -version, --version</code>	Instructs to output information on the product and engine versions and exit Dr.Web Scanner .
<code>-ki</code>	Instructs to output information about the license and its owner (in UTF8 encoding only).
<code>-go</code>	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.
<code>-ot</code>	Instructs to use the standard output (<code>stdout</code>).
<code>-oq</code>	Disables information output.
<code>-ok</code>	Instructs to list all scanned objects in the report and mark the "clean" object with Ok .
<code>-log=[+] <path to file></code>	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.
<code>-ini=<path to file></code>	Instructs to use the specified configuration file. By default, Dr.Web Scanner uses <code>drweb32.ini</code> (this configuration file is shared by Dr.Web Daemon , Dr.Web Scanner and Dr.Web Updater). Dr.Web Scanner uses parameters specified in the <code>[Scanner]</code> section of this file. The list of the scanner parameters and available values are similar to the those specified in the <code>[Daemon]</code> section .
<code>-lng=<path to file></code>	Instructs to use the specified language file. The default language is English.
<code>-a = <Control Agent address></code>	Run Dr.Web Scanner in the central protection mode.
<code>-ni</code>	Disables the use of the configuration file for adjusting scanner settings. Dr.Web Scanner is configured via command line parameters.
<code>-ns</code>	Disables interruption of scanning process even upon receipt of interruption signals (SIGINT).
<code>--only-key</code>	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 UNIX mail servers** configuration files organization, see [Configuration files](#).

[Scanner]

EnginePath = {path to file}	Location of <code>drweb32.dll</code> module (anti-virus engine Dr.Web Engine). This parameter is also used by Dr.Web Updater . <u>Default value:</u> EnginePath = <code>%bin_dir/lib/drweb32.dll</code>
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 <code>.vdb</code> extension <u>Default value:</u> VirusBase = <code>%var_dir/bases/*.vdb</code>
UpdatePath = {path to directory}	This parameter is used by Dr.Web Updater (<code>update.pl</code>) and is mandatory. <u>Default value:</u> UpdatePath = <code>%var_dir/updates/</code>
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 <u>Default value:</u> TempPath = <code>/tmp/</code>



LngFileName = {path to file}	Language file location. By default, language files have a .dwl extension <u>Default value:</u> 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 <u>Default value:</u> Key = %bin_dir/drweb32.key
OutputMode = {Terminal Quiet}	Output mode: <ul style="list-style-type: none">• Terminal – console output• Quiet – no output <u>Default value:</u> 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. <u>Default value:</u> 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). <u>Default value:</u> ScanPriority = 0
FileTypes = {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. <u>Default value:</u> FileTypes = 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
FileTypesWarnings = {logical}	Notifies about files of unknown types. <u>Default value:</u> FileTypesWarnings = Yes
ScanFiles = {All ByType}	Instructs to scan all files (All value) or only files with the extensions specified in the FileType parameter (ByType value).



	<p>The parameter can have the <code>ByType</code> value only in the local scan mode. In other modes, the value must be set to <code>All</code>.</p> <p>All mail fails are scanned regardless of the <code>scanFiles</code> parameter value.</p> <p>Default value: ScanFiles = All</p>
ScanSubDirectories = {logical}	<p>Enables or disables scanning of subdirectories.</p> <p>Default value: ScanSubDirectories = Yes</p>
CheckArchives = {logical}	<p>Enables or disables checking of files in archives (RAR, ARJ, TAR, GZIP, CAB and others).</p> <p>Default value: CheckArchives = Yes</p>
CheckEMailFiles = {logical}	<p>Enables or disables checking mail files.</p> <p>Default value: CheckEMailFiles = Yes</p>
ExcludePaths = {list of path file masks}	<p>Masks for files to be skipped during scanning. Multiple values are allowed (separated by commas).</p> <p>Default value: ExcludePaths = /proc,/sys,/dev</p>
FollowLinks = {logical}	<p>Allows or forbids Dr.Web Scanner to follow symbolic links during scanning.</p> <p>Default value: FollowLinks = No</p>
RenameFilesTo = {mask}	<p>Mask for renaming files when the <code>Rename</code> action is applied.</p> <p>Default value: RenameFilesTo = #??</p>
MoveFilesTo = {path to directory}	<p>Path to the Quarantine directory.</p> <p>Default value: MoveFilesTo = %var_dir/infected/</p>
EnableDeleteArchiveAction = {logical}	<p>Enables or disables <code>Delete</code> action for complex objects (archives, mailboxes, HTML pages) if they contain infected files.</p> <p>Please note, if the action is enabled, a whole complex object is to be deleted. Use this option carefully!</p> <p>Default value: EnableDeleteArchiveAction = No</p>
InfectedFiles = {action}	<p>Sets one of the following actions upon detection of an infected file: Report, Cure, Delete, Move, Rename, Ignore.</p> <p>Delete and Move actions are applied to a whole complex object upon detection of infected files within it.</p> <p>Default value: InfectedFiles = Report</p>



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



<code>ActionInfectedContainer = {action}</code>	<p>Sets one of the following actions upon detection of an infected file in a container (OLE, HTML, PowerPoint, etc.):</p> <p>Report, Delete, Move, Rename, Ignore.</p> <p><u>Default value:</u> ActionInfectedContainer = Report</p> <p>Logging parameters:</p>
<code>LogFileName = {syslog file name}</code>	<p>Log file name.</p> <p>You can specify <code>syslog</code> as a log file name to use <code>syslogd</code> system service for logging.</p> <p>In this case you must also specify the SyslogFacility and SyslogPriority parameters.</p> <p><u>Default value:</u> LogFileName = <code>syslog</code></p>
<code>SyslogFacility = {syslog label}</code>	<p>Log type label which is used by <code>syslogd</code> system service.</p> <p><u>Default value:</u> SyslogFacility = <code>Daemon</code></p>
<code>SyslogPriority = {log level}</code>	<p>Log verbosity level when <code>syslogd</code> system service is used.</p> <p>The following levels are allowed:</p> <ul style="list-style-type: none">• Error• Alert• Warning• Info• Notice <p><u>Default value:</u> SyslogPriority = <code>Info</code></p>
<code>LimitLog = {logical}</code>	<p>Enables or disables limit of log file size (if LogFileName value is not set to <code>syslog</code>).</p> <p>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.</p> <p><u>Default value:</u> LimitLog = <code>No</code></p>
<code>MaxLogSize = {numerical value}</code>	<p>Maximum log file size in Kbytes.</p> <p>Used only with LimitLog = <code>Yes</code>.</p> <p>If this parameter value is set to 0, log file size is not checked.</p> <p><u>Default value:</u> MaxLogSize = 512</p>
<code>LogScanned = {logical}</code>	<p>Enables or disables logging of information about all scanned objects regardless whether they are infected or not.</p> <p><u>Default value:</u> LogScanned = <code>Yes</code></p>
<code>LogPacked = {logical}</code>	<p>Enables or disables logging of additional information about files packed with DIET, PKLITE and other utilities.</p>



	<p><u>Default value:</u> LogPacked = Yes</p>
LogArchived = {logical}	<p>Enables or disables logging of additional information about files archived with various archiving utilities.</p> <p><u>Default value:</u> LogArchived = Yes</p>
LogTime = {logical}	<p>Enables or disables logging of time for each record. Parameter is not used if LogFileNames = syslog.</p> <p><u>Default value:</u> LogTime = Yes</p>
LogStatistics = {logical}	<p>Enables or disables logging of scan statistics.</p> <p><u>Default value:</u> LogStatistics = Yes</p>
RecodeNonprintable = {logical}	<p>Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters).</p> <p><u>Default value:</u> RecodeNonprintable = Yes</p>
RecodeMode = {Replace QuotedPrintable}	<p>Decoding mode for non printable characters if RecodeNonprintable = Yes.</p> <p>When RecodeMode = Replace, all non-printable characters are substituted with the RecodeChar parameter value (see below).</p> <p>When RecodeMode = QuotedPrintable, all non-printable characters are converted to the Quoted Printable encoding.</p> <p><u>Default value:</u> RecodeMode = QuotedPrintable</p>
RecodeChar = {"?" "_" ...}	<p>Sets character for replacing non-printable characters if RecodeMode = Replace.</p> <p><u>Default value:</u> RecodeChar = "?"</p>
<p>The following parameters can be used to reduce time of scanning archives (by skipping some objects in an archive).</p>	
MaxCompressionRatio = {numerical value}	<p>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".</p> <p>Parameter can have only natural values.</p> <p>If the value is set to 0, compression ratio will not be checked</p> <p><u>Default value:</u> MaxCompressionRatio = 5000</p>
CompressionCheckThreshold = {numerical value}	<p>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).</p>



	<p>Default value: CompressionCheckThreshold = 1024</p>
MaxFileSizeToExtract = {numerical value}	<p>Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped.</p> <p>An email message with such a file is considered as a "mail bomb".</p> <p>Default value: MaxFileSizeToExtract = 500000</p>
MaxArchiveLevel = {numerical value}	<p>Maximum archive nesting level.</p> <p>If an archive nesting level exceeds the specified value, the archive is skipped.</p> <p>An email message with such a file is considered as a "mail bomb".</p> <p>If the value is set to 0, archive nesting level will not be checked</p> <p>Default value: MaxArchiveLevel = 8</p>
MaximumMemoryAllocationSize = {numerical value}	<p>Maximum size of the memory (in Mbytes) that can be used by Dr.Web Scanner to check one file.</p> <p>If the value is set to 0, memory allocation is not limited.</p> <p>Default value: MaximumMemoryAllocationSize = 0</p>
ScannerScanTimeout = {numerical value}	<p>Maximum time period allowed for scanning one file (in seconds).</p> <p>If the value is set to 0, scanning time is not limited.</p> <p>Default value: ScannerScanTimeout = 0</p>
MaxBasesObsolescencePeriod = {numerical value}	<p>Maximum time (in hours) after last update when virus databases are considered as up-to-date.</p> <p>Upon the expiration of this time period, notification displays informing that the databases are obsolete.</p> <p>If the value is set to 0, database actuality will not be checked.</p> <p>Default value: MaxBasesObsolescencePeriod = 24</p>
ControlAgent = {address}	<p>Dr.Web Agent socket address.</p> <p>Example:</p> <pre>ControlAgent = inet:4040@127.0.0.1,local:%var_dir/ipc/.agent</pre> <p>Dr.Web Scanner receives a license key file and configuration from Dr.Web Agent. (if OnlyKey = No).</p> <p>Default value: ControlAgent = local:%var_dir/ipc/.agent</p>
OnlyKey = {logical}	<p>Enables receiving only a license key file from Dr.Web Agent, without configuration. At that, Dr.Web Scanner uses the local configuration file.</p> <p>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).</p>



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).



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 UNIX mail servers** is a ready-made solution for integrating **Dr.Web Daemon** with UNIX mail servers..



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	
<u>Description:</u> Show information about supported command line parameters on the screen and terminate the module		
-a		<Agent socket address>
<u>Description:</u> Start Dr.Web Daemon in the central protection mode under control of the specified copy of Dr.Web Agent		
-ini		<path to file>
<u>Description:</u> Module must use the specified configuration file		
	--foreground	<yes no>
<u>Description:</u> Operation mode of Dr.Web Daemon . If <code>yes</code> is specified, Dr.Web Daemon is a foreground process. Otherwise (<code>no</code>), Dr.Web Daemon is a background process		
	--check-only	<command line parameters for checking>
<u>Description:</u> Check Dr.Web Daemon configuration correctness on startup. If any command line parameter is specified, correctness of the value is also checked		
	--only-key	
<u>Description:</u> 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 `LogFile` 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.d11`). 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 UNIX mail servers** 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.

Usage of **Dr.Web Daemon** with mail servers requires special attention, because mail filters usually work as the mail system and use its privileges. In the local scan mode, a mail filter usually creates a file with the message received from the mail system and provides **Dr.Web Daemon** a path to it. At this



step, carefully specify access permissions to the directory for filters to create appropriate files. We recommend either to include a user whose privileges are used by **Dr.Web Daemon** into the mail subsystem group or run **Dr.Web Daemon** with the privileges of the mail system user.



If the system is configured correctly, **Dr.Web Daemon** does not require `root` superuser privileges..

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 UNIX mail servers** 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 `LogFile` 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 UNIX mail servers**) in the `ClientsLogs` parameter value.

Regardless of the `ClientsLogs` parameter, if **Dr.Web Daemon** recognizes its client, scanning results will be 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 <code>drweb32.dll</code> module (anti-virus engine Dr.Web Engine). This parameter is also used by the Dr.Web Updater . Default value: EnginePath = <code>%bin_dir/lib/drweb32.dll</code>
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 <code>.vdb</code> extension Default value: VirusBase = <code>%var_dir/bases/*.vdb</code>
UpdatePath =	Directory to store updates. The parameter is mandatory.



	<p>Default value:</p> <p>UpdatePath = %var_dir/updates/</p>
<p>TempPath = {path to directory}</p>	<p>Directory where the Dr.Web Engine anti-virus engine puts temporary files.</p> <p>It is used when system has insufficient memory or to unpack certain types of archives.</p> <p>Default value:</p> <p>TempPath = %var_dir/spool/</p>
<p>Key = {path to file}</p>	<p>Key file location (license or demo). By default, a key file has the .key extension.</p> <p>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.</p> <p>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.</p> <p>Default value:</p> <p>Key = %bin_dir/drweb32.key</p>
<p>MailAddressesList = {path to file}</p>	<p>This parameter is used only if you have an email license for less than 50 addresses.</p> <p>Specified file must contain a list of email addresses (no more than specified by the license, one email address per line), for which both incoming and outgoing messages will be checked. Aliases are treated as separate addresses.</p> <p>Default value:</p> <p>MailAddressesList = %etc_dir/email.ini</p>
<p>OutputMode = {Terminal Quiet}</p>	<p>Output mode:</p> <ul style="list-style-type: none">• Terminal - console output• Quiet - no output <p>Default value:</p> <p>OutputMode = Terminal</p>
<p>RunForeground = {logical}</p>	<p>Allows to disable or enable daemon mode for Dr.Web Daemon.</p> <p>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).</p> <p>Default value:</p> <p>RunForeground = No</p>
<p>User = {text value}</p>	<p>User under which Dr.Web Daemon operates.</p> <p>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.</p> <p>This parameter cannot be changed when reloading configuration using SIGHUP.</p>



	<p>Default value:</p> <p>User = drweb</p>
<p>PidFile = {path to file}</p>	<p>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).</p> <p>If more than one Socket parameter is specified, this file contains information on all the sockets (one per line).</p> <p>This file is created every time Dr.Web Daemon starts.</p> <p>Default value:</p> <p>PidFile = %var_dir/run/drwebd.pid</p>
<p>BusyFile = {path to file}</p>	<p>File where Dr.Web Daemon busy flag is stored.</p> <p>This file is created by a Dr.Web Daemon child process upon receipt of the scan command and is removed after successful command execution.</p> <p>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).</p> <p>Default value:</p> <p>BusyFile = %var_dir/run/drwebd.bsy</p>
<p>ProcessesPool = {process pool settings}</p>	<p>Settings of dynamic process pool.</p> <p>At first, specify the number of processes in the pool:</p> <ul style="list-style-type: none">• 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. <p>Then specify optional secondary parameters:</p> <ul style="list-style-type: none">• 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. <p>Default value:</p> <p>ProcessesPool = auto, timeout = 120, stat = no, stop_timeout = 1</p>
<p>OnlyKey = {logical}</p>	<p>Enables receiving only a license key file from Dr.Web Agent, without configuration. At that, Dr.Web Scanner uses the local configuration file.</p> <p>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).</p>



	<p>Default value: OnlyKey = No</p>
ControlAgent = {address}	<p>Dr.Web Agent socket address.</p> <p>Example: ControlAgent = inet:4040@127.0.0.1,local:%var_dir/ipc/.agent</p> <p>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).</p> <p>Default value: ControlAgent = local:%var_dir/ipc/.agent</p>
MailCommand = {string}	<p>Shell command used by Dr.Web Daemon and Dr.Web Updater for sending notifications on new updates to the user (administrator) via email.</p> <p>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.</p> <p>Default value: MailCommand = "/usr/sbin/sendmail -i -bm -f drweb -- root"</p>
NotifyPeriod = {numerical value}	<p>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.</p> <p>If the parameter value is set to 0, Dr.Web Daemon starts sending out notifications immediately after the key file expires.</p> <p>Default value: NotifyPeriod = 14</p>
NotifyFile = {path to file}	<p>Path to the file with a timestamp of the last license expiration notification.</p> <p>Default value: NotifyFile = %var_dir/.notify</p>
NotifyType = {Ever Everyday Once}	<p>Frequency of sending license expiration notifications.</p> <ul style="list-style-type: none">• 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. <p>Default value: NotifyType = Ever</p>
FileTimeout = {numerical value}	<p>Maximum time (in seconds) allowed for Dr.Web Daemon to perform scanning of one file.</p> <p>If the parameter value is set to 0, time to scan of one file is unlimited.</p> <p>Default value: FileTimeout = 30</p>
StopOnFirstInfected = {logical}	<p>Enables or disables interruption of file scanning upon detection of</p>



	<p>the first virus.</p> <p>If the value is set to <code>yes</code>, it can significantly reduce mail server load and scan time.</p> <p><u>Default value:</u> StopOnFirstInfected = No</p>
ScanPriority = {signed numerical value}	<p>Priority of Dr.Web Daemon process.</p> <p>Value must be in the following range: -20 (highest priority) to 19 (lowest priority for Linux) or 20 (lowest priority for FreeBSD and Solaris).</p> <p><u>Default value:</u> ScanPriority = 0</p>
FileTypes = {list of file extensions}	<p>Types of files to be checked "by type", that is, when the ScanFiles parameter value (described below) is set to <code>ByType</code>.</p> <p>"*" and "?" wildcard characters are allowed.</p> <p><u>Default value:</u> FileTypes = 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</p>
FileTypesWarnings = {logical}	<p>Notify on files of unknown types</p> <p><u>Default value:</u> FileTypesWarnings = Yes</p>
ScanFiles = {All ByType}	<p>Scan only files with extensions specified in the FileTypes parameter (the <code>ByType</code> value) or all files (the <code>All</code> value).</p> <p>This parameter can have the <code>ByType</code> value only in the local scan mode (in other modes, only the <code>All</code> value can be set).</p> <p>In mailboxes, all files are always checked (regardless of the ScanFiles parameter value).</p> <p>Attention! If the ScanType value is set to <code>local</code> or <code>auto</code> for the Drweb anti-virus plug-in, setting the <code>ByType</code> value to the ScanFiles parameter will force this plug-in to skip all email messages WITHOUT anti-virus check!</p> <p><u>Default value:</u> ScanFiles = All</p>
CheckArchives = {logical}	<p>Enables or disables checking of files in archives.</p> <p>The following formats are supported: ZIP (WinZip, InfoZIP, etc.), RAR, ARJ, TAR, GZIP, CAB and others.</p> <p><u>Default value:</u> CheckArchives = Yes</p>
CheckEMailFiles = {logical}	<p>Enables or disables checking of email files.</p> <p><u>Default value:</u> CheckEMailFiles = Yes</p>



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 <code>Rename</code> 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 <code>syslog</code> as a log file name and logging will be performed by <code>syslogd</code> system service. In this case, also specify the <code>SyslogFacility</code> and <code>SyslogPriority</code> parameter values. Default value: LogFileName = syslog
SyslogFacility = {syslog label}	Log type label used by <code>syslogd</code> system service. Default value: SyslogFacility = Daemon
SyslogPriority = {log level}	Logging priority (log verbosity level) when <code>syslogd</code> system service is used. There are the following levels allowed: <ul style="list-style-type: none">• Error• Alert• Warning• Info• Notice Default value: SyslogPriority = Info
LimitLog = {logical}	Enables or disables limit for log file size (if <code>LogFileName</code> value is not specified to <code>syslog</code>). If limit is enabled, Dr.Web Daemon checks the size of a log file on startup or on receipt of <code>HUP</code> signal. If the log file size is greater than <code>MaxLogSize</code> value, the log file is overwritten with an empty file and logging starts from scratch.



	<p><u>Default value:</u> LimitLog = No</p>
MaxLogSize = {numerical value}	<p>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.</p> <p><u>Default value:</u> MaxLogSize = 512</p>
LogScanned = {logical}	<p>Enables or disables logging of information about all scanned objects regardless whether they are infected or not.</p> <p><u>Default value:</u> LogScanned = Yes</p>
LogPacked = {logical}	<p>Enables or disables logging of additional information about files packed with DIET, PKLITE and other utilities.</p> <p><u>Default value:</u> LogPacked = Yes</p>
LogArchived = {logical}	<p>Enables or disables logging of additional information about files archived with various archiving utilities.</p> <p><u>Default value:</u> LogArchived = Yes</p>
LogTime = {logical}	<p>Enables or disables logging of time for each record. The parameter is not used if LogFileName = syslog.</p> <p><u>Default value:</u> LogTime = Yes</p>
LogProcessInfo = {logical}	<p>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.</p> <p><u>Default value:</u> LogProcessInfo = Yes</p>
RecodeNonprintable = {logical}	<p>Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters).</p> <p><u>Default value:</u> RecodeNonprintable = Yes</p>
RecodeMode = {Replace QuotedPrintable}	<p>Decoding mode for non-printable characters (if RecodeNonprintable = Yes).</p> <p>When RecodeMode = Replace, all non-printable characters are substituted with the RecodeChar parameter value (see below).</p> <p>When RecodeMode = QuotedPrintable, all non-printable characters are converted to Quoted Printable encoding.</p> <p><u>Default value:</u> RecodeMode = QuotedPrintable</p>



<pre>RecodeChar = { "?" "_" ... }</pre>	<p>Sets a character to replace all non-printable characters if RecodeMode = Replace.</p> <p><u>Default value:</u> RecodeChar = "?"</p>
<pre>Socket = { address list }</pre>	<p>List of sockets to be used for communication with Dr.Web Daemon (separated by commas).</p> <p>Example: <code>Socket = inet:3000@127.0.0.1,local:%var_dir/.daemon</code></p> <p>You can also specify a socket address in the following format: PORT [interfaces] FILE [access].</p> <p>For a TCP socket, specify a decimal port number (PORT) and the list of interface names or IP addresses for incoming requests (interfaces).</p> <p>Example: <code>Socket = 3000 127.0.0.1, 192.168.0.100</code></p> <p>For UNIX sockets, specify a socket name (FILE) and access permissions in the octal form.</p> <p>Example: <code>Socket = %var_dir/.daemon 0660</code></p> <p>Number of Socket parameter values is not limited. Dr.Web Daemon will work with all sockets described correctly.</p> <p>To enable connections on all available interfaces, set 3000 0.0.0.0 as a value of this parameter.</p> <p><u>Default value:</u> Socket = %var_dir/run/.daemon</p>
<pre>SocketTimeout = { numerical value }</pre>	<p>Maximum time (in seconds) allowed for transferring data through socket (file scanning time is not included).</p> <p>If the parameter value is set to 0, the time is unlimited.</p> <p><u>Default value:</u> SocketTimeout = 10</p>
<pre>ClientsLogs = { string list }</pre>	<p>Enables splitting of log files.</p> <p>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.</p> <p>If more than six values are set, the configuration file is considered invalid.</p> <p>Log files are defined in the following way: <client name1>:<path to file>, <client name2>:<path to file></p> <p>Client name may be one of the following:</p> <ul style="list-style-type: none">• 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;



	<ul style="list-style-type: none"> lotus — Dr.Web for IBM Lotus Domino. <p>Example: drwebdc:/var/drweb/log/drwebdc.log, smb:syslog, mail:/var/drweb/log/drwebmail.log</p> <p><u>Default value:</u></p>
<p>MaxBasesObsolescencePeriod = {numerical value}</p>	<p>Period, in hours, after last update, during which virus databases are considered up-to-date.</p> <p>When this period is over, a message notifying that databases are obsolete is output.</p> <p>If value is set to 0, database obsolescence is not checked.</p> <p><u>Default value:</u> MaxBasesObsolescencePeriod = 24</p>

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.

<p>MaxCompressionRatio = {numerical value}</p>	<p>Maximum compression ratio, that is a ratio between size of unpacked file and its size within an archive.</p> <p>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.</p> <p>Value of this parameter must be not less than 2.</p> <p><u>Default value:</u> MaxCompressionRatio = 5000</p>
<p>CompressionCheckThreshold = {numerical value}</p>	<p>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.</p> <p><u>Default value:</u> CompressionCheckThreshold = 1024</p>
<p>MaxFileSizeToExtract = {numerical value}</p>	<p>Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped.</p> <p><u>Default value:</u> MaxFileSizeToExtract = 40960</p>
<p>MaxArchiveLevel = {numerical value}</p>	<p>Maximum allowed archive nesting level.</p> <p>If an archive nesting level exceeds the specified value, an archive is not scanned.</p> <p><u>Default value:</u> MaxArchiveLevel = 8</p>
<p>MessagePatternFileName = {path to file}</p>	<p>Path to template for a license expiration message.</p>



	<p>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:</p> <ul style="list-style-type: none">• \$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. <p>If there is no user-defined template, standard message in English is output.</p> <p>Default value: MessagePatternFileName = %etc_dir/templates/drwebd/msg.tmpl</p>
<p>MailTo = {email address}</p>	<p>Email address of an administrator where the following information is sent: messages about license expiration, virus databases obsolescence, etc.</p> <p>Default value: MailTo =</p>



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 UNIX mail servers**. **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 UNIX mail servers** executable files.

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

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 UNIX mail servers** requires regular updates to virus databases.

Dr.Web for UNIX mail servers 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 UNIX mail servers** 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: `dwrXYYYY.vdb` and `dwnXYYYY.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;
- `drwXXXXYY.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;
- `dwnXXXXYY.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;
- `dwrXXXXYY.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

For Linux: 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.

For FreeBSD and Solaris: manual configuration of `cron` is required to enable its interaction with **Dr.Web Updater**.

For example, when you use **FreeBSD** you may add the following string to `crontab` of `drweb` user:

```
*/30 * * * * /usr/local/drweb/update.pl
```

If you work with **Solaris**, the following set of commands is used:

```
# crontab -e drweb
# 0,30 * * * * /opt/drweb/update.pl
```

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 recommended 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 UNIX mail servers**) 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 UNIX mail servers** 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 by 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 UNIX mail servers**.

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 UNIX mail servers** 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}	<p>If Yes value is specified, Dr.Web Updater does not update Dr.Web Daemon and Dr.Web Scanner. It updates only the plug-ins.</p> <p><u>Default value:</u> UpdatePluginsOnly = No</p>
Section = {Daemon Scanner}	<p>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: <code>Scanner</code>, <code>Daemon</code>.</p> <p>Value of this parameter can be temporarily overridden by the <code>--what</code> command line parameter. The specified value is used until the next start of the script.</p> <p><u>Default value:</u> Section = <code>Daemon</code></p>
ProgramPath = {path to file}	<p>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.</p> <p><u>Default value:</u> ProgramPath = <code>%bin_dir/drwebd</code></p>
SignedReader = {path to file}	<p>Path to the program which is used to read digitally signed files.</p> <p><u>Default value:</u> SignedReader = <code>%bin_dir/read_signed</code></p>
LzmaDecoderPath = {path to directory}	<p>Path to the directory that contains a program used for unpacking of lzma-archives.</p> <p><u>Default value:</u> LzmaDecoderPath = <code>%bin_dir/</code></p>
LockFile = {path to file}	<p>Path to the file used to prevent sharing of certain files during their processing by Dr.Web Updater.</p> <p><u>Default value:</u> LockFile = <code>%var_dir/run/update.lock</code></p>
CronSummary = {logical}	<p>If you specify Yes, Dr.Web Updater outputs an update report for each session to <code>stdout</code>.</p> <p>This mode can be used to send notifications to administrator by email, if Dr.Web Updater is run by the <code>cron</code> daemon.</p> <p><u>Default value:</u> CronSummary = <code>Yes</code></p>
DrlFile = {path to file}	<p>Path to the file (<code>*.drl</code>) with the list of Dr.Web GUS servers.</p> <p>Dr.Web Updater selects a server from this list in random order to download updates.</p> <p>For details on downloading updates, see Updating Process.</p> <p>This file is signed by Doctor Web and must not be modified by a user. The file is updated automatically.</p> <p><u>Default value:</u> DrlFile = <code>%var_dir/bases/update.drl</code></p>



CustomDrlFile = {path to file}	<p>Path to the file (*.drl) with the alternative list of Dr.Web GUS servers.</p> <p>Dr.Web Updater also selects a server from this list in random order to download updates.</p> <p>For details on downloading updates, see Updating Process.</p> <p>This file is signed by Doctor Web and must not be modified by a user. It is updated automatically.</p> <p>Default value: CustomDrlFile = %var_dir/bases/custom.drl</p>
FallbackToDrl = {logical}	<p>Allows using the file specified by DrlFile when connection to one of the servers listed in CustomDrlFile failed.</p> <p>If the parameter value is No, the file specified in DrlFile is not used.</p> <p>If the file specified in CustomDrlFile does not exist, the file specified in DrlFile is used regardless of the FallbackToDrl parameter value.</p> <p>For details on downloading updates, see Updating Process.</p> <p>Default value: FallbackToDrl = Yes</p>
DrlDir = {path to directory}	<p>Path to the directory that contains drl files with lists of Dr.Web GUS servers for each plug-in.</p> <p>These files are signed by Doctor Web and must not be modified by a user.</p> <p>Default value: DrlDir = %var_dir/drl/</p>
Timeout = {numerical value}	<p>Maximum wait time for downloading updates from the selected Dr.Web GUS server, in seconds.</p> <p>Default value: Timeout = 90</p>
Tries = {numerical value}	<p>Number of attempts by Dr.Web Updater to establish connection with the selected update server.</p> <p>Default value: Tries = 3</p>
ProxyServer = {host name IP address}	<p>Host name or IP address of the proxy server which is used for Internet access.</p> <p>If the proxy server is not used, the value of this parameter must be empty.</p> <p>Default value: ProxyServer =</p>
ProxyLogin = {string}	<p>User login to access the used proxy server (if it requires authentication).</p> <p>Default value: ProxyLogin =</p>



ProxyPassword = {string}	The password to access the used proxy server (if it requires authentication). <u>Default value:</u> ProxyPassword =
LogFileName = {syslog file name}	Path to the log file name. You can specify <code>syslog</code> as a log file name and logging will be performed by <code>syslogd</code> system service. <u>Default value:</u> LogFileName = <code>syslog</code>
SyslogFacility = {syslog label}	Log type label which is used by <code>syslogd</code> system service. <u>Default value:</u> SyslogFacility = <code>Daemon</code>
LogLevel = {log level}	Log verbosity level . The following levels are allowed: <ul style="list-style-type: none">• Quiet• Error• Warning• Info• Debug• Verbose <u>Default value:</u> LogLevel = <code>Info</code>
MaildPidFile = {path to file}	Path to PID file of Dr.Web MailD . <u>Default value:</u> MaildPidFile = <code>%var_dir/run/drweb-maild.pid</code>
BlacklistPath = {path to directory}	Path to the directory with <code>.dws</code> files. <u>Default value:</u> BlacklistPath = <code>%var_dir/dws</code>
AgentConfPath = {path to file}	Path to Dr.Web Agent configuration file. <u>Default value:</u> AgentConfPath = <code>%var_dir/agent.conf</code>
PathToVadeRetro = {path to file}	Path to the <code>libvaderetro.so</code> library (used by Vaderetro plugin). <u>Default value:</u> PathToVadeRetro = <code>%var_dir/lib/libvaderetro.so</code>
ExpiredTimeLimit = {numerical value}	Number of days left before license expiration during which Dr.Web Updater is attempting to update license key file. <u>Default value:</u>



	<code>ExpiredTimeLimit = 14</code>
<code>ESLockfile =</code> {path to file}	Path to the lock file. If the lock file exists, Dr.Web Updater can not be automatically initialized by <code>cron</code> daemon.
	Default value: <code>ESLockfile = %var_dir/run/es_updater.lock</code>

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 **Dr1File** and **CustomDr1File** 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 **FallbackToDr1** parameter;
 - If both files are accessible, **Dr.Web Updater** uses the file specified in the **CustomDr1File** parameter;
 - If it is impossible to connect to any of the servers from this file (specified in **CustomDr1File**), and the **FallbackToDr1** value is set to `Yes`, **Dr.Web Updater** tries to establish connection with the servers from the file specified in the **Dr1File** 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 Agent

Dr.Web Agent is a resident module used to manage settings of **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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.

It can be convenient to set mail processing rules in the **Dr.Web MailD** configuration file: instead of writing one big rule, you can split it into several separate rules.

Example:

```
GlobalRules = select message, append_html "lookup:file:/maild-files/
somehtml.html"
```

This rule can also be specified in the following way:

```
GlobalRules = select message
GlobalRules = append_html "lookup:file:/maild-files/somehtml.html"
```

(see description of **Dr.Web Modifier** rules format).

Please note that splitting of a line into several parts with backslashes allows you to insert comments which are ignored by the rule parser when reading configuration.

Example:

```
to:user@host cont \
modifier/LocalRules=select mime.headers "X-Spam-Level" "\\*\\*\\*\\*", \
# 3 and more asterisks <- this is a comment (is ignored)
if found,\
select mime.headers Subject ".*",\
replace "[SPAM]" "^",\
endif
```

(for details on the rule format, see the [Rules] section description).



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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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**.

If settings of modules for interaction with MTA (`drweb-courier`, `drweb-cgp-receiver`) were updated on the central protection server, send a `SIGHUP` or `STOP-START` signal to the modules for the update to take effect.

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 UNIX mail servers** are unlocked and restored to their previous or default values. You can access all features of **Dr.Web for UNIX mail servers** solutions again and configure them.



For correct operation in the standalone mode, **Dr.Web for UNIX mail servers** 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 UNIX mail servers** and **Dr.Web Anti-virus for Linux** together in the central protection mode

Because of the implementation features, **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers**, 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
<code>-h</code>	<code>--help</code>	



Short case	Extended case	Arguments
<u>Description:</u> Show information about supported command line parameters on the screen and terminate the module		
-v	--version	
<u>Description:</u> Show Dr.Web Agent version on the screen and terminate the module		
-u	--update-all	
<u>Description:</u> Start updating all Dr.Web for UNIX mail servers components		
-f	--update-failed	
<u>Description:</u> Start updating Dr.Web for UNIX mail servers components, updating of which failed in the standard mode		
-C	--check-only	
<u>Description:</u> 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>
<u>Description:</u> Enable the module to use the specified configuration file		
-d	--droppwd	
<u>Description:</u> 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	
<u>Description:</u> Change username and password required to access Dr.Web Enterprise Server		
-s	--socket	<path to file>
<u>Description:</u> Use the specified socket for interaction with the controlled modules		
-P	--pid-file	<path to file>
<u>Description:</u> Use the specified file as a PID file of Dr.Web Agent		
-e	--export-config	<application name>
<u>Description:</u> 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 UNIX mail servers** 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: <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug <u>Default value:</u> Level = Info
IPCLevel = {log level}	Log verbosity level of IPC library. The following levels are available: <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug <u>Default value:</u> IPCLevel = Error
SyslogFacility = {syslog label}	Log type label used by <code>syslogd</code> system service. <u>Default value:</u> SyslogFacility = Daemon
FileName = {path to file syslog}	Path to the log file. You can specify <code>syslog</code> as a log file name and logging will be performed by <code>syslogd</code> system service. <u>Default value:</u> FileName = <code>syslog</code>

[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. <u>Default value:</u> MetaConfigDir = <code>%etc_dir/agent/</code>
---	--



UseMonitor = {logical}	Yes value indicates to drweb-agent that Dr.Web Monitor is used as a part of Dr.Web for UNIX mail servers . 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 UNIX mail servers** 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 : <ul style="list-style-type: none"> • 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 : <ul style="list-style-type: none"> • 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.
	<u>Default value:</u> 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.
	<u>Default value:</u> StatisticsServer = stat.drweb.com:80/update
StatisticsUpdatePeriod = {numerical value}	Period (in minutes) for statistics updating. Value cannot be less than 5
	<u>Default value:</u> 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 <code>http_proxy</code> environment variable is used.
	<u>Example:</u> <code>StatisticsProxy = localhost:3128</code>
	<u>Default value:</u> StatisticsProxy =
StatisticsProxyAuth = {text value}	Authentication string (<username>:<password>) to access proxy server.
	<u>Example:</u> <code>StatisticsProxyAuth = test:testpwd</code>
	<u>Default value:</u> 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).
	<u>Default value:</u> 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 UNIX mail servers** update via **Dr.Web Enterprise Server**:

[Update]

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

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 UNIX mail servers** 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 `MetaConfigDir` 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 UNIX mail servers** 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 from Dr.Web MailD package for Linux:

```
Application "MAILD"
id 40
ConfFile "/etc/drweb/mailed_smtp.conf"
Components
lookup_ldap LDAP
lookup_regex REGEX
drweb-mailed General, Logging, MailBase, Stat, Mailed, Filters,
    Quarantine, /_Rules=Rule*:Rules, /Reports/Send,
    /Reports/SendTime, /Reports/Names, /Reports/MaxPoolSize,
    /Reports/MaxStoreInDbPeriod, Reports/CheckForRemovePeriod,
    /Notifier/FilterMail, /Notifier/NotifyLangs,
    /Notifier/LngBaseDir
drweb-notifier General, Logging, Notifier, /Sender/Method, /_Rules,
    Reports, /Filters/BeforeQueueFilters,
    /Filters/AfterQueueFilters, /Quarantine/AccessByEmail,
    /Quarantine/StoredTime
drweb-sender General, Logging, Sender
drweb-receiver General, Logging, /Mailed/ProtectedNetworks,
    /Mailed/ProtectedDomains, /Mailed/IncludeSubdomains,
    SASL, Receiver
EndComponents
EndApplication
```

Integration with Dr.Web Enterprise Security Suite

There are two possible situations which require integration of **Dr.Web for UNIX mail servers** with **Dr.Web Enterprise Security Suite**:

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

To enable **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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`, `MAILD`).

Example:

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

Starting the System

To start the system:

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

For **Linux** and **Solaris**:

```
# /etc/init.d/drweb-monitor start
```

For **FreeBSD**:

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

Integration with Dr.Web ESS 10

Dr.Web for UNIX mail servers 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.



The products, operating in **FreeBSD** 6.x, cannot be integrated with **Dr.Web ESS** 10.

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 UNIX mail servers** 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 in **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,MAILD
```

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

3. If required, configure parameters in `es_daemon.conf` file that is used by **Dr.Web Daemon** respectively in **enterprise** mode.
4. 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.
5. 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 UNIX mail servers** 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 16. 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>
  <dwvolid>62083</dwvolid>
  <place>1</place>
  <percents>34.201062139103</percents>
</item>
<item>
  <vname>Win32.HLLM.MyDoom</vname>
  <dwvolid>9353</dwvolid>
  <place>2</place>
  <percents>25.1303270912579</percents>
</item>
<item>
  <vname>Win32.HLLM.Beagle</vname>
  <dwvolid>26997</dwvolid>
  <place>3</place>
  <percents>13.4593034783378</percents>
</item>
<item>
  <vname>Trojan.Botnetlog.9</vname>
  <dwvolid>438003</dwvolid>
  <place>4</place>
  <percents>7.86446592583328</percents>
</item>
<item>
  <vname>Trojan.DownLoad.36339</vname>
  <dwvolid>435637</dwvolid>
  <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 Monitor

Dr.Web Monitor is a memory resident module `drweb-monitor`.

It is used to increase fault-tolerance of the whole **Dr.Web for UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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 UNIX mail servers** 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.



- 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 UNIX mail servers** are unlocked and restored to their previous or default values. You can access all settings of **Dr.Web for UNIX mail servers** again and configure them.



For correct operation in the standalone mode, **Dr.Web for UNIX mail servers** 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	
<u>Description:</u> Show information about supported command line parameters on the screen and terminate the module		
-v	--version	
<u>Description:</u> Show Dr.Web Monitor version on the screen and terminate the module		
-u	--update	
<u>Description:</u> Start updating all Dr.Web for UNIX mail servers components		
-C	--check-only	
<u>Description:</u> 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>
<u>Description:</u> Check correctness of configuration of all Dr.Web for UNIX mail servers components		
-c	--conf	<path to file>
<u>Description:</u> Module must use the specified configuration file		
-r	--run	<application name>[,<application name>,...]
<u>Description:</u> 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, MAILD
```

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 UNIX mail servers** 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]

<pre>Level = {log level}</pre>	<p>Dr.Web Monitor log verbosity level.</p> <p>The following levels are available:</p> <ul style="list-style-type: none"> • Quiet • Error • Alert • Info • Debug <p><u>Default value:</u> Level = Info</p>
<pre>IPCLevel = {log level}</pre>	<p><u>Log verbosity level</u> for IPC library.</p> <p>The following levels are available:</p> <ul style="list-style-type: none"> • Quiet • Error • Alert • Info • Debug <p><u>Default value:</u> IPCLevel = Error</p>
<pre>SyslogFacility = {syslog label}</pre>	<p><u>Log type label</u> which is used by syslogd system service.</p> <p><u>Default value:</u> SyslogFacility = Daemon</p>
<pre>FileName = {syslog path to file}</pre>	<p>Path to the log file.</p> <p>You can specify <code>syslog</code> as a log file name and logging will be performed by <code>syslogd</code> system service. In this case, you must also specify the SyslogFacility parameter.</p> <p><u>Default value:</u> FileName = <code>syslog</code></p>

[Monitor] Section

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

[Monitor]

<pre>RunForeground = {logical}</pre>	<p>Yes value forbids Dr.Web Monitor to operate in daemon mode.</p> <p>This option can be used by some monitoring utilities (for example, daemontools).</p>
--------------------------------------	---



	<p>Default value: RunForeground = No</p>
User = {text value}	<p>Name of the user whose privileges are used by Dr.Web Monitor.</p> <p>Please note that when Dr.Web MailD solution operates in SMTP \LMTP proxy mode or integrated with CGP MTA or Exim MTA, value of this parameter must be set to <code>root</code>.</p> <p>Default value: User = drweb</p>
Group = {text value}	<p>User group name used to run Dr.Web Monitor with certain user privileges.</p> <p>Please note that when Dr.Web MailD solution operates in SMTP \LMTP proxy mode or integrated with CGP MTA or Exim MTA, value of this parameter must be set to <code>root</code>.</p> <p>Default value: Group = drweb</p>
PidFileDir = {path to directory}	<p>Path to the directory of a file where information on Dr.Web Monitor process identifier (PID) is written upon the module startup.</p> <p>Default value: PidFileDir = %var_dir/run/</p>
ChDir = {path to directory}	<p>Change of working directory upon Dr.Web Monitor startup.</p> <p>If this parameter is set, Dr.Web Monitor changes directory to the one specified in this parameter value. Otherwise, working directory is not changed.</p> <p>Default value: ChDir = /</p>
MetaConfigDir = {path to directory}	<p>Path to the directory where metaconfiguration files reside.</p> <p>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.</p> <p>Default value: MetaConfigDir = %etc_dir/monitor/</p>
Address = {address}	<p>Socket used by Dr.Web Monitor to receive control signals from other Dr.Web components.</p> <p>Default value: Address = local:%var_dir/ipc/.monitor</p>
Timeout = {numerical value}	<p>Maximum time (in seconds) to establish connection between Dr.Web Monitor and other Dr.Web components.</p> <p>Default value: Timeout = 5</p>
TmpFileFmt = {text value}	<p>Name templates for Dr.Web Monitor temporary files.</p> <p>Template format: <code>path_to_file.XXXXXX</code></p> <p>where x is a random symbol (letter or digit), used in temporary file names.</p>



	<p>Default value: TmpFileFmt = %var_dir/messages/tmp/monitor.XXXXXX</p>
RunAppList = {text value}	<p>List of modules started by Dr.Web Monitor; use comma as a delimiter.</p> <p>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.</p> <p>Default value: RunAppList = AGENT</p>
UseEnterpriseMode = {logical}	<p>If the value is set to <i>Yes</i>, Dr.Web Monitor receives the list of modules to be started from Dr.Web Agent rather than from the RunAppList parameter value.</p> <p>Default value: UseEnterpriseMode = No</p>
RecoveryTimeList = {numerical values}	<p>Time intervals between attempts to restart components that are not responding (in seconds).</p> <p>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.</p> <p>Default value: RecoveryTimeList = 0,30,60</p>
InjectCmd = {string}	<p>Command to send reports.</p> <p>Please note that if you want to send reports to other addresses (not only to <i>root@localhost</i>), you need to specify the addresses in the command.</p> <p>Default value: InjectCmd = "/usr/sbin/sendmail -t"</p>
AgentAddress = {address}	<p>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).</p> <p>Default value: AgentAddress = local:%var_dir/ipc/.agent</p>
AgentResponseTime = {numerical value}	<p>Maximum time to wait a response from <i>drweb-agent</i> module in seconds.</p> <p>If Dr.Web Agent does not respond during this time period, Dr.Web Monitor considers <i>drweb-agent</i> not working and tries to restart it.</p> <p>If 0 is specified, response time is unlimited.</p> <p>Default value: AgentResponseTime = 5</p>



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 UNIX mail servers** 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 (for **Linux** and **Solaris**);
- add `drweb_monitor_enable="YES"` line to the `/etc/rc.conf` file (for **FreeBSD**).



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 UNIX mail servers** 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 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 UNIX mail servers**



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 drweb:drweb
EndComponents
EndApplication
```

Example of mmc file for Dr.Web MailD:

```
Application "MAILD"
FullName "Dr.Web (R) MailD"
Path "/opt/drweb/"
Depends "AGENT"
Components
# name args MaxStartTime MaxStopTime NotifyType User:Group
drweb-notifier local:/var/drweb/ipc/.agent 30 30 MAIL drweb:drweb
drweb-sender local:/var/drweb/ipc/.agent 15 30 LOG drweb:drweb
drweb-maild local:/var/drweb/ipc/.agent 120 30 MAIL drweb:drweb
drweb-receiver local:/var/drweb/ipc/.agent 15 30 MAIL root:drweb
EndComponents
EndApplication
```



Dr.Web MailD

Dr.Web MailD is a group of interacting software modules. These modules execute receiving, processing (scan for viruses and spam) and resending (to end user MUA or MTA) of email messages. **Dr.Web MailD** can operate as:

- a proxy server for SMTP and LMTP protocols;
- filter for supported mail transfer systems. Integration with the following MTA is supported:
 - **Sendmail**;
 - **Postfix**;
 - **Exim**;
 - **CommuniGate Pro**;
 - **Courier**;
 - **Zmailer**;
 - **Qmail**.
- a **proxy server for post protocols of end-user MUA** (POP3/IMAP), which operates as a filter between MTA and MUA.

Parameters of **Dr.Web MailD** operation mode and separate modules are defined in the [configuration files](#). Also it is possible to control **Dr.Web MailD** via the special interactive [management interface](#) like a command line.

Structure of Dr.Web MailD

General structure of **Dr.Web for UNIX mail servers** is described in the [Introduction](#).

Structure of **Dr.Web MailD** is presented in the figure below. Also this figure shows the scheme of interaction between **Dr.Web MailD** and MTAs.

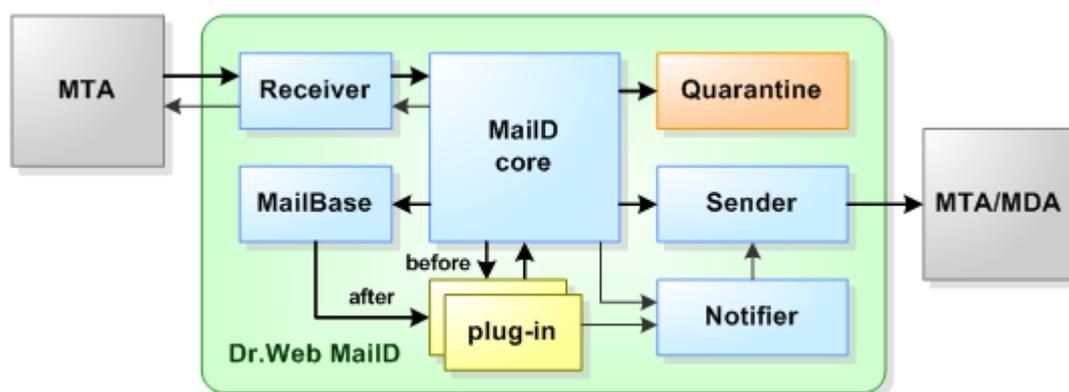


Figure 17. Structure of Dr.Web MailD

Short description of **Dr.Web MailD** is given in the following table.

Component	Description
Receiver	<p>The component is used to receive email messages from mail systems (MTA or MDA). Component functions are performed by different modules (<code>drweb-receiver</code>, <code>drweb-milter</code>, <code>drweb-cgp-receiver</code>, and others) that are included in the solution structure depending on the post system integrated with Dr.Web for UNIX mail servers.</p> <p>The component supports simultaneous work as part of several modules that perform Receiver functions, which allows receiving and processing email messages from several sources at once.</p>



Component	Description
	Some modules also support modification/dispatching of received email messages as well as receiving of verification results from the MailD core component (for example, the drweb-milter module supports this option and returns verification results to the Sendmail mail system before the SMTP session terminates).
MailBase	<p>Special database which stores received email messages until their check completes and they are sent to the recipient if the plug-ins operate in the asynchronous mode. email messages are stored as files in file storage.</p> <p>The component also contains lists of users (message recipients), their groups, their personal settings of message processing as well as aggregated statistics.</p>
MailD core	<p>The main component. Parses the MIME parts of email messages, transmits messages for check to plug-ins and saves messages into a database. Received verification results are sent either to the Receiver component (if the message is processed in the asynchronous mode and the timeout is not expired) or to the Sender component. The checked message, if allowed, is transferred for delivery to the Sender component.</p> <p>Functions of this component are performed by the drweb-maild module.</p>
Quarantine	The special directory of a file system or DBI storage. Used for temporary storage of email messages which were marked as suspicious or malicious (if an email message was not deleted or rejected).
Sender	<p>This component sends to MTA or MDA processed email messages including notifications, reports and DSN. Messages are sent either directly to the mail system or to mail servers (via SMTP/LMTP protocol). Sender functions also can be performed by different executable modules (drweb-sender, drweb-cgp-sender and others, depending on the mail system integrated with Dr.Web for UNIX mail servers). Usually the used Sender components is paired with the used Receiver component.</p> <p>Sender can receive and process requests from Maild core, Notifier and Monitor for sending messages and notifications.</p>
Notifier	<p>The component creates special (service) email messages of two main types:</p> <ul style="list-style-type: none">• MailD Notifications – email message sent from the address specified in the FilterMail parameter value. The message contains information on Dr.Web MailD operation (for example, on virus detection in a received message);• DSN (<i>delivery status notification</i>) – automatic email message created by MTA. The message contains information on any problem occurred during message processing. This message always has an empty FROM header (as required). <p>All notifications are created during Dr.Web MailD operation. Requests for notifications can be received either from a plug-in (for example, if a virus is found in a message) or from other components (if configured accordingly). For example, MailD core can send a request for creation of general statistic report and Sender can send a request for creation of DSN about failure to dispatch an email message.</p> <p>MailD notifications can be sent to senders and recipients of email messages and to the system administrator.</p> <p>Functions of this component are performed by the drweb-notifier module.</p>
plugin	<p>Additional modules used for analysis and check of email messages (for viruses, suspicious software and spam attributes). Messages are processed by plug-ins in a user-defined order. List of used plug-ins can be changed at any time without restart of Dr.Web MailD. To do this, change the corresponding parameters in the configuration file and send SIGHUP signal to MailD core or Dr.Web Monitor.</p> <p>General set of plug-ins (concrete set depends on the installed distribution kit):</p> <ul style="list-style-type: none">• Drweb – plug-in for anti-virus check of email messages. It uses Dr.Web Daemon that receives email messages that are already parsed.• Dr.Web HeadersFilter – plug-in for filtering of email messages by their headers. The filtering rules can be set using Perl regular expressions.• Vaderetro – plug-in which checks mail for spam messages. It uses a VadeRetro library that is regularly updated and provides spam recognition of high quality.



Component	Description
	<ul style="list-style-type: none">• Dr.Web Modifier – plug-in allows modification of an email message (or its any part) according to some conditions. For example, the plug-in allows adding a text label to check email messages or deleting pictures from messages marked as spam.

Concrete [module set](#) is defined by the installed **Dr.Web for UNIX mail servers** distribution kit and [integrated mail system](#).

Dr.Web MailD features

Dr.Web MailD uses a developed system of [Message processing Rules](#) that allows for advanced control of

- email messages processing and their internal routing;
- statistics generating;
- sending notifications (MailD notifications and DSN).

For mail processing rules and all [Lookup](#) configuration parameters, you can use data retrieved from text files and relational databases using LDAP. The following database management systems (DBMS) are supported:

- **Oracle**;
- **MySQL**;
- **PostgreSQL**;
- **SQLite**;
- **Firebird**;
- **CDB**;
- **Berkeley**.

Dr.Web MailD can also use **ODBC** API to connect to any data source via the corresponding **ODBC** driver.

Work with each data source is made independently of others, with the use of connection settings, specified for this source only. Mail processing rules and parameters of the `Lookup` type can simultaneously use data retrieved from different sources.

Dr.Web MailD can also organize and use built-in database of mail users and their groups, where a mail user is a sender or recipient of a processed email message. In the built-in database, you can specify mail processing rules for a specific user (or a user group), as well as rules for individual or group statistics gathering.



Operations with the built-in database are performed only via [Interactive Management Interface](#).

To prevent spam distribution and DHA attacks, **Dr.Web MailD** features special technologies: one that determines [reputation of a client's IP addresses](#), [score management](#) (of both connection and message scores), and [SMTP restriction management](#) that allows to filter a suspicious client at the connecting and mail transfer stage. These technologies improve effectiveness of spam filtering and reduce load on the protected mail system.

When an email message is quarantined, the sender (or recipient) can manage it using special [control email messages](#). Control messages can be sent from the user's MUA in response to notifications on moving the message to **Quarantine**.

If volume and intensity of mail traffic is high, **Dr.Web MailD** allows organizing solution work in the [cluster mode](#). In this mode, the program components are located on several servers with load distribution in order to increase performance. To provide users with this option, special **proxy modules** (**Proxy client** and **Proxy server**) are included in the solution. The **proxy modules** provide transparent remote interaction of the **Sender** and **Receiver** components with **MailD core**.



If required, interaction with several MTA/MDA can be provided due to the [mechanism of simultaneous work](#) of paired **Sender** and **Receiver** components (at that, the components can be of [different types](#)).

Message Processing

Algorithm of mail processing

1. **Receiver** gets email messages from MTA and transmits them to the **MailD core** component which is responsible for message check.
2. **MailD core** performs MIME-parsing of the messages and transmits them to [plug-ins](#). The components is also responsible for saving the messages to the storage.

Parameters for the processed message are selected in the following order:

- a) search for [Rules](#) stored in the [built-in database](#) and associated with the **recipient** (the recipient is determined according to the RCPT TO specified by the sender).
- b) search for [Rules](#) stored in the [built-in database](#) and associated with the user groups which the recipient belongs to. The search is performed in reverse order until the parameter value is found: from the last to the first group in the list.
- c) search for the parameter value from Rules defined in the [configuration file](#) ([Rules] [section](#)).

Rules are searched according to the following procedure:

- All Rules are checked in the order they are specified in the current group of Rules.
- For each Rule, its `CONDITION` is checked. If the `CONDITION` is true, the parameter value is searched in the `SETTINGS` part of the Rule.
- If the `CONDITION` is false, the Rule is rejected and the next Rule is checked.
- If the `CONDITION` is true and is followed by the `cont` directive, the Rule is rejected and the next Rule is checked. If the `CONDITION` is true and is followed by the `stop` directive, the search stops regardless of whether the parameter value is found or not.

As the result of a rule check, the parameter value is determined as follows:

- When the searched parameter is found in one of the Rules which `CONDITION` is true, the value is taken from the `SETTINGS` part of this Rule (if more than one Rule are true, the result parameter value depends on its semantics. For details, see [Rules of message processing](#)).
- If no Rule is specified, no Rule is true or no Rule with true `CONDITION` contains value for the required parameter, its value is taken from the corresponding section of the configuration file.
- If the configuration file does not contain value for the required parameter, its default value is used.



Note that if an email message has several recipients, truth of the rule is checked for each of them. For details, see [Rules of message processing](#).

3. Plug-ins can process email messages in the **synchronous mode 'before-queue'** (if the corresponding plug-ins are specified) and in the **asynchronous mode 'after-queue'** after saving messages to the storage (if the corresponding plug-ins are specified).
4. Results of email message check are transmitted to **Receiver** (if the processing results are not timed out).
5. If an email message was regarded as malicious or suspicious, it can be rejected, deleted or moved to **Quarantine**. If the message was rejected, notifications to the sender and, if required, to the recipient are sent by the **Notifier** component.
6. **Sender** dispatches all outgoing email messages to external mail servers or mail systems. It can send checked email messages as well as notifications and reports generated by **Dr.Web MailD** modules.



Processing modes

Dr.Web MailD uses two modes of message processing:

1. **Synchronous mode ("before-queue")**: message, received from the sender with the **Receiver** component, is processed by plug-ins "on the fly", without sending the message to the **MailBase** storage. In this mode, messages are processed by the plug-ins specified in the **BeforeQueue** [list](#). The **Receiver** component does not send a response to the sender until processing ends or time-out is expired. If the message is regarded as malicious or suspicious, the sender gets the error code in response from the **Receiver**.
2. **Asynchronous mode ("after-queue")**: Message, received from the sender by the **Receiver** component is saved to the **MailBase** storage before the message processing starts. Then **Receiver** responds to the sender with notification that the message was correctly accepted. After that, the message is processed by plug-ins specified in the **AfterQueue** [list](#). If the message is regarded as malicious or suspicious, the sender receives DSN with the report on the check results.

If some plug-ins are specified in the **BeforeQueue** list, and others - in the **AfterQueue** list, messages are processed in the synchronous mode and then in the asynchronous mode.

Note that if **Dr.Web HeadersFilter** and **Dr.Web Modifier** plug-ins are used (that is, if local rules of message processing or Rules that override their parameter values are specified), it is recommended to assign the plug-ins to the **AfterQueueFilters** [queue](#) if **Dr.Web MailD** operates as the [SMTP/LMTP proxy](#). If **Dr.Web MailD** is [integrated](#) with any MTA, assign all plug-ins to the **BeforeQueueFilters** [queue](#), but increase the IPC timeout value (**IPCTimeout** parameter from the [\[General\]](#) [section](#) in the configuration file). If these plug-ins are not used for message processing, it is recommended to remove them from all of the queues in the [\[Filters\]](#) [section](#).

In both modes, if a message is not rejected (i.e., if `discard`, `reject` or `tempfail` [actions](#) were not applied to it), it is transmitted to the **Sender** component for delivery to the recipient. When transmitted in the synchronous mode, the message is also synchronously sent, that is, **Receiver** waits either for sending results from the **Sender** component or time-out occurrence. Time-out is used to ensure the correct interaction with external MTAs.



Please note that best modes for **Dr.Web MailD** operation depend on:

- type and intensity of processed mail traffic
- MTA integrated with **Dr.Web MailD**
- method of interaction with the integrated MTA.

Thus, before you change the default settings, please, read the description of integration with the selected MTA in the [corresponding](#) part of this Manual.

Features of interaction between Receiver, MailD core and Sender in different modes

1. Time limit of **Receiver** and **MailD core** interaction is restricted to the value of the **IPCTimeout** parameter (defined in the [\[General\]](#) [section](#) of the configuration file). When operating in the synchronous mode, this parameter also restricts time limit for interaction between the **MailD core** and **Sender** components. The `drweb-milter` [module](#) uses the maximum value out of those set for the **IPCTimeout** and **ProcessingTimeout** parameters in [\[Milter\]](#) [section](#).
2. When operating in the **asynchronous mode**, the received email message is saved to the internal queue of **Dr.Web MailD**. After that, **Receiver** immediately responds with the 250 SMTP code indicating that the email message is queued. **Dr.Web MailD** transmits the message to **Sender** which dispatches it further. In case of channel latency or if the component cannot connect to the target mail server, **Sender** delays dispatching of the message.

In this mode, time limit, restricted by the **IPCTimeout** parameter value, must be commensurate with the average time of message processing by all of the [plug-ins](#).

3. When operating in the **synchronous mode**, **Receiver** does not respond to the sender until the



email message is processed by all of the plug-ins and dispatched further by **Sender**. If the time exceeds the value of `IPCTimeout` – 1 second (`drweb-milter` uses the greater value out of the `IPCTimeout` and `ProcessingTimeout` parameters) and **Sender** did not dispatch the message, the component skips all attempts to connect to the target MTA (if more than one connection attempt is specified for **Sender** in the `Address` or `Router` parameters in the [Sender] [section](#)). After that, **Sender** delays dispatching of the message. In this case, **Receiver** sends the SMTP response `250 Maild Error` that indicates queuing of the email message that is to be sent as soon as problems with MTA connection are fixed. The "Maild Error" message indicates that the situation is abnormal for the synchronous mode (it is supposed to process mail traffic and transfer it to the target MTA rapidly). Also in this case errors of the "ERROR Broken pipe" type may be [logged](#). These errors indicate that **Sender** was trying to send to **Maild core** a report through the connection closed upon the time-out.

General recommendations

1. Synchronous mode is **not supposed for intense workload**. It is recommended to select this mode only when **Dr.Web MailD** operates as a local mail filter. **Do not select this mode** if **Dr.Web MailD** operates as a high load SMTP gateway.
2. If most part of mail traffic through **Dr.Web MailD** consists of messages with large attachments (or with large number of attachments), as well as if delays in the channel or problems with the target MTA are possible, it is recommended to select the asynchronous mode. Also, it is not recommended to assign plug-ins to the **before-queue** if those plug-ins can process a message during a long time period.
3. When the synchronous mode is selected, it is strongly recommended not to decrease the `IPCTimeout` default value. If it is necessary to decrease the value, ensure that it is commensurate with the time required for processing by all of the plug-ins (`IPCTimeout` value must be greater). If the time-out occurs during processing of a message, it can be lost and not delivered to the recipient. In this mode, it is also assumed that no delays occur when **Sender** dispatches messages to the target MTA (it must be available and correctly configured).

Features of operation with MTA via Milter protocol:

1. If connection between **Dr.Web MailD** and MTA is established via the `Milter` protocol (the `drweb-milter` [module](#) is used as **Receiver**), returning of a checked message back to the MTA queue is configured by the `CanChangeBody` parameter value (specified in the [Milter] [section](#) of the configuration file).
2. If some plug-ins are assigned to the `BeforeQueue` (that is, the **synchronous** mode is selected), the parameter `CanChangeBody = Yes`, and significant delays occur while message processing (for example, one of the plug-ins does not finish message processing during the period specified by `ProcessingTimeout`), then MTA receives SMTP response code specified in the `ProcessingError` parameter (all mentioned parameters are specified in the [Milter] [section](#) of the configuration file).
3. When the **synchronous** mode is selected, the parameter `CanChangeBody = No` and a component (**Maild core** or **Sender**) stops responding while processing a message, MTA receives SMTP `451` code in response.
4. If some plug-ins are assigned to `AfterQueue` (i.e., the **asynchronous** mode is selected), the parameter `CanChangeBody = Yes` and significant delays occur when a message is processed (for example, one of the plug-ins does not finish message processing during the period specified by `ProcessingTimeout`), then MTA receives SMTP `250 queued` code in response, but the message is still processed and sent via the **Sender** module (`drweb-sender` plug-in). Thus, if the parameter `CanChangeBody = Yes`, the asynchronous mode is not recommended as it provides neither speed gain (additional time for saving service files to the storage is required), nor reliability gain.
5. When operating in the **asynchronous** mode, `CanChangeBody = No`, and a component (**Maild core** or **Sender**) is not responding during message processing, the message is lost and MTA



receives SMTP 250 queued in response. Thus, this **setting combination is strongly not recommended!**

- Thus, it is recommended not to select the asynchronous mode (i.e., assigning plug-ins to **AfterQueue**) when connecting to MTA via the **Milter** protocol, as this mode does not provide any gain in message processing.

Moreover, it is recommended to [optimize operation and system resources usage](#) if the following problems occur: delays in message processing, queues when receiving or sending messages, not responding errors, shortage of system resources.

Used Modules

The following table describes the plug-ins that are included into the **Dr.Web MailD** mail processing component:

Plug-in	Component	Description
Mandatory plug-ins		
<code>drweb-maild</code>	MailD core	Central module that provides operation of Dr.Web MailD
<code>drweb-notifier</code>	Notifier	Module that creates notifications, reports and DSN
<code>drweb-receiver</code>	Receiver (SMTP/LMTP)	Module interacting with MTA (receiving of incoming messages) via the SMTP/LMTP protocol
<code>drweb-sender</code>	Sender (SMTP/LMTP)	Module interacting with MTA (sending of outgoing messages) via the SMTP/LMTP protocol As <code>drweb-sender</code> can transfer messages directly to the local post system, the module is used almost in all schemes of Dr.Web MailD and MTA integration.
<code>drweb-proxy-client</code>	Proxy client	Client proxy module for Dr.Web MailD clustering The module used as a stub to a cluster node responsible for receiving and sending messages (i.e., where Sender and Receiver are installed)
<code>drweb-proxy-server</code>	Proxy server	Server proxy module for Dr.Web MailD clustering . Used as a stub to a cluster node responsible for mail processing (i.e., where the MailD core central component is installed as well as the built-in MailDB database and plug-ins)
<code>drweb-imap</code>	IMAP filter	MDA proxy module via the IMAP protocol (check of messages when transferring them from MDA to MUA)
<code>drweb-pop3</code>	POP3 filter	MDA proxy module via the POP3 protocol (check of messages when transferring them from MDA to MUA)
Optional plug-ins (used for integration with specific post systems; the plug-ins can be not required)		
<code>drweb-zmailer</code>	Receiver (Zmailer)	Module for integration with the Zmailer post system (receiving messages that are to be processed)
<code>drweb-qmail</code>	Receiver (Qmail)	Module for integration with the Qmail post system (receiving messages that are to be processed)
<code>drweb-courier</code>	Receiver (Courier)	Module for integration with the Courier post system (receiving messages that are to be processed)
<code>drweb-cgp-sender</code>	Sender (CommuniGate Pro)	Module for integration with the CommuniGate Pro post system (dispatching processed messages)
<code>drweb-cgp-receiver</code>	Receiver (CommuniGate Pro)	Module for integration with the CommuniGate Pro post system (receiving messages that are to be processed)



Plug-in	Component	Description
<code>drweb-milter</code>	Receiver (Milter)	Module for integration with post systems via the Milter protocol
Utilities		
<code>drweb-inject</code>		Utility for force dispatching of messages
<code>drweb-lookup</code>		Utility for Lookup validity check
<code>drweb-qcontrol</code>		Utility for Quarantine management
<code>drweb-qp</code>		Utility for Quarantine management (meant for interaction with DBI). The utility cannot be started manually

All modules described in the table above are located in the `%bin_dir` directory.

Command Line Parameters

Like every UNIX program, **Dr.Web MailD** modules support command line parameters. The command line format is as follows:

```
<module_name> [parameters] <agent_socket>
```

where:

- `<module_name>` - name of the module;
- `<parameters>` - optional command line parameters;
- `<agent_socket>` - socket used for receiving module configuration from **Dr.Web Agent component**.

General Parameters

In the current version of **Dr.Web for UNIX mail servers**, the modules support the following command line parameters:

Short case	Extended case	Arguments
<code>-h</code>	<code>--help</code>	
<u>Description:</u> Show information about supported command line parameters on the screen and exit		
<code>-v</code>	<code>--version</code>	
<u>Description:</u> Show module version on the screen and exit		
<code>-l</code>	<code>--level</code>	<code><log verbosity level></code>
<u>Description:</u> Verbosity level for logging information on module startup (default value is <code>info</code>)		
<code>-t</code>	<code>--timeout</code>	<code><number of seconds></code>
<u>Description:</u> Maximum wait time for receiving configuration from Dr.Web Agent		
	<code>--component</code>	<code><components name></code>
<u>Description:</u> Name of the component that requests configuration from Dr.Web Agent		
Note that this command line parameter is not supported by <code>drweb-zmailer</code>!		
	<code>--log-name</code>	<code><components name></code>
<u>Description:</u> Name of the component that performs logging		



Short case	Extended case	Arguments
	--check-only	
<p><u>Description:</u> Start the component in the configuration check mode. To provide correct operation, Dr.Web Agent must be previously started. If configuration test is successful, the following message is output to the console: "Options OK". If configuration test failed, the following message is output: "Options ERROR".</p> <p><u>Note that this command line parameter is not supported by drweb-zmailer!</u></p>		

Example:

```
drweb-maild -t 30 local:%var_dir/ipc/.agent
```

This command starts **Dr.Web MailD** with 30 seconds time-out for receiving configuration from **Dr.Web Agent** via the local:%var_dir/ipc/.agent socket.

Module Specific Parameters

Different modules of **Dr.Web MailD** also support additional command line parameters that are specific for a certain module. The exceptions are **drweb-notifier** and **drweb-proxy-client** modules that can be used only with the parameters mentioned above.

1. drweb-maild

The following command line parameters are specific for **drweb-maild** and are used to check correctness of [message processing Rules](#):

Short case	Extended case	Arguments
-s	--sender	<mail address>
<u>Description:</u> Sender's address (from the message envelope)		
-r	--recipient	<mail address>
<u>Description:</u> Recipient address (from the message envelope). To set several recipients, specify the parameter several times		
-b	--block	<object name>
<u>Description:</u> Name of the blocking object found in the message (for example, virus name). To set several blocking objects, specify the parameter several times		
	--client-ip	<IP address>
<u>Description:</u> IP address of the client which sent the message		
	--server-ip	<IP address>
<u>Description:</u> IP address of the server interface that received the message		
	--client-port	<port number>
<u>Description:</u> Port of a client from which the message was received		
	--server-port	<port number>
<u>Description:</u> Port of a server which received the message		
	--server-us	<UNIX socket>
<u>Description:</u> Name of the UNIX socket of the server that received the message		



Short case	Extended case	Arguments
	--id	<identifier>
<u>Description:</u> Unique identifier of Receiver that transmitted the message		
	--auth	
<u>Description:</u> Flag indicating that the message was received from an authorized user		
	--size	<size>
<u>Description:</u> Size of the checked message (the value is of the size type)		
	--score	<score>
<u>Description:</u> Score assigned to the message (number)		
	--md-client	<MailDesk Client name>
<u>Description:</u> Unique identifier of the MailDesk Client		

2. Other plug-ins (except for **drweb-zmailer**) have two specific parameters used for [operation with several](#) **Sender** and **Receiver** components simultaneously:

Short case	Extended case	Arguments
	--unique-id	<identifier>
<u>Description:</u> Unique component identifier. This parameter enables MailD core to work with several copies of Sender and Receiver components. Every new Sender+Receiver pair must be started with the unique identifier (paired Sender and Receiver components must have the same ID). Thus, an email message must be dispatched by Sender which has the same ID as Receiver which got the message. If the component with the required ID is not found, the default Sender is selected.		
List of available Senders is reinitialized via the SIGHUP signal.		
The following list shows the way MailD core treats a component ID for each of the modules:		
<ul style="list-style-type: none"> • <code>drweb-receiver</code> - Receiver's identifier; • <code>drweb-sender</code> - Sender's identifier; • <code>drweb-proxy-server</code> - Identifier shared by paired Receiver and Sender interacting with MailD core via this component (see Using Internal Proxy); • <code>drweb-imap</code> - IMAP filter's identifier (used similar to the Receiver identifier for search of the corresponding Sender); • <code>drweb-pop3</code> - POP3 filter's identifier (used similar to the Receiver identifier for search of the corresponding Sender); • <code>drweb-milter</code> - Receiver's identifier; • <code>drweb-cgp-receiver</code> - Receiver's identifier; • <code>drweb-cgp-sender</code> - Sender's identifier; • <code>drweb-courier</code> - Receiver's identifier; • <code>drweb-qmail</code> - Receiver's identifier. 		
	--section	<ИМЯ СЕКЦИИ>
<u>Description:</u> Name of the configuration file section that contains settings for the module. If the parameter is not specified, the module uses settings from the default section.		
These are the following default sections for each module:		
<ul style="list-style-type: none"> • <code>drweb-receiver</code> - [Receiver] • <code>drweb-sender</code> - [Sender] • <code>drweb-proxy-server</code> - [ProxyServer] • <code>drweb-imap</code> - [IMAP] 		



Short case	Extended case	Arguments
• drweb-pop3	- [POP3]	
• drweb-milter	- [Milter]	
• drweb-cgp-receiver	- [CgpReceiver]	
• drweb-cgp-sender	- [CgpSender]	
• drweb-courier	- [Courier]	
• drweb-qmail	- [Qmail]	

3. drweb-zmailer

The following command line parameters are specific for **drweb-zmailer**:

Short case	Extended case	Arguments
-u	--user	<user name>
<u>Description:</u> User account name under which drweb-maild is running.		
Note that if the parameter is not specified, drweb-zmailer is running with <code>root</code> rights, which can cause interoperability problems with drweb-maild if it is not running with <code>root</code> rights.		
-i	--ipcllevel	<verbosity level>
<u>Description:</u> Log verbosity level for IPC library used by drweb-zmailer .		
Available values: quiet, error, alert, info, debug		
-f	--facility	<syslog flag>
<u>Description:</u> Used syslog label (if the syslog service is enabled).		
Available values: daemon, mail, local0, ..., local7		
-b	--basedir	<directory path>
<u>Description:</u> Path to the home directory where Dr.Web MailD plug-ins are located		
	--id	<identifier>
<u>Description:</u> Similar to the <code>--unique-id</code> parameter specified for other components (see above). MailD core treats this parameter as a Receiver identifier.		
	--log-filename	<log file name>
<u>Description:</u> Name of the used log file or <code>syslog</code> (if the syslog service is enabled)		
	--file	<path to the file>
<u>Description:</u> Path to the file which must be processed on the plug-in startup		
	--hash	<value>
<u>Description:</u> The SecureHash parameter value from the [Sender] section of Dr.Web MailD configuration file		
	--interface	<0 1>
<u>Description:</u> Version of the used smtpserver : 0 – for 2.99.55 version or earlier, 1 – for 2.99.56 version or later		
-e	--error-action	<action>
<u>Description:</u> Action applied if an internal error occurs in the plug-in during message processing.		



Short case	Extended case	Arguments
Available values: pass, reject, discard, tempfail		
-Z		<file path>
Description: Path to the ZMailer configuration file that is to be ignored		

For information on utility command line parameters, refer to the description of the [corresponding utility](#) in the present document.

Processed Signals

All program modules constantly residing in memory support processing of the following signals:

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

Some modules can process additional signals:

- Upon the receipt of `SIGUSR2` signal, **Sender** component, makes an attempt to send all messages from the internal queue.
- Upon the receipt of `SIGUSR1` signal, `drweb-receiver` module saves statistics on **SMTP restrictions** to the `restriction.txt` file.
- Upon the receipt of `SIGUSR1` signal, all components save files with statistics on operation of dynamic thread pools and persistent connections.

Files with statistics are saved to the directory specified in the `BaseDir` parameter from the [General] section of **Dr.Web MailD** configuration file. Statistics on operation of dynamic thread pools is logged if the `Debug` verbosity level is specified. Thus, if the specified verbosity level is less detailed than `Debug`, statistics is not logged.

For details on **Dr.Web MailD** internal statistics format, see **Internal Statistics** section.



Please note that not all parameters can change their values upon receipt of `SIGHUP` signal (that is, if you change values of these parameters and send `SIGHUP` signal to **Dr.Web MailD** modules, changes will not be applied). That is designated in the description of such parameters. If it is necessary to change their values, restart **Dr.Web MailD**.

Also note that both **Dr.Web Monitor component** and **Dr.Web Agent component** do not support processing of `SIGUSR1` and `SIGUSR2` signals. Upon receipt of these signals, the components terminate.



Logging

Logged data can be output to the file or to the **syslog** system service. Output destination is defined in the configuration file of **Dr.Web MailD**, in the [Logging] [section](#).

When **syslog** is used, every string looks as follows:

```
'['tid']' name[.sub] level [sid(/mta-id)] text
```

where:

- `tid` - identifier of the thread which is responsible for string output;
- `name` - name of the component which performs an output (for example, [plug-in](#) or [module](#) name);
- `sub` - name of the component service to perform output.

The most important services are:

- `ipc` - interprocess communication service;
- `thrN` - support service of the [thread pool](#) with the N number;
- `report` - report support service;
- `ldap`, `odbc`, `oracle`, `sqlite`, `mysql`, `postgres`, `cdb`, `berkeley`, `firebird` - support service of the [corresponding](#) Lookups;
- `control` - [interactive management](#) service;
- `parser` - [notification templates](#) parser service;
- `MRS` - service of receiving messages via SMTP/LMTP;
- `smtp` - service of sending messages via SMTP;
- `lmtp` - service of sending messages via LMTP;
- `pipe` - service of sending messages through the pipe;
- `queue` - service of processing internal queue.
- `level` - [log verbosity level](#). The following values are available: FATAL, ERROR, WARN, INFO, DEBUG.
- `sid` - session identifier of a message to which that log line is related. The number must be specified in hexadecimal notation;
- `mta-id` - identifier of a message inside MTA, from which this message is received. It is output only if **Dr.Web MailD** is not operating in **SMTP/LMTP proxy** mode and the identifier was received from MTA;
- `text` - text of a log message.



When any module is started, its [log verbosity level](#) is set to INFO by default. After configuration is received from **Dr.Web Agent**, this level is changed according to the [specified settings](#). To enable DEBUG logging during module initialization (for example, to gain information on parameters received from **Dr.Web Agent**), use the `--level` [command line parameter](#) (set its value to debug).

Internal Statistics

Dr.Web MailD can collect internal statistics of the following kinds:

1. Statistics on **SMTP restrictions** appliance
2. Statistics on operation of dynamic thread pools and connections.



It is recommended to gather internal statistics on **Dr.Web MailD** operation to estimate the load and its bandwidth. This information can be used to [optimize operation and system resources usage](#).

Statistics on SMTP restrictions appliance

Statistics on appliance of restrictions on different SMTP session stages is saved to the `restrictions.txt` file. Restrictions for **Receiver** are specified in the `[Receiver]` [section](#) of the **Dr.Web MailD** configuration file. Statistics is saved only if the value of the `RestrictionStat` parameter in the same is set to `Yes`.

Statistics records are always added to the end of file. Each 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 the previous statistics record
- `curr` – time of the current statistics record
- `period` – time interval between the previous and current records.

Further, after an empty line, applied SMTP restrictions are listed (one line for one restriction). For example:

```
reject_unknown_domain: total: 19   trusted: 0   reject: 0   tempfail: 0
```

For each restriction, the following information is output:

- restriction type;
- total count of messages for which this restriction was checked (`total`);
- count of messages which were marked as `trusted` upon applying of this restriction (`trusted`);
- count of messages which were rejected upon applying of this restriction (`reject`);
- count of messages which senders received `Tempfail` SMTP status upon applying of this restriction (`tempfail`).

Please note that statistics is not aggregated. Each saved record contains statistics on applied restrictions between previous and current moments of saving. Statistics on restriction appliance is collected only if **Dr.Web MailD** is operating in **SMTP/LMTP proxy** mode.

Statistics on operation of thread pools and connections

Statistics on operation of thread pools and connections is collected if the corresponding option for the pool is enabled (for example, the `InPoolOptions` and `OutPoolOptions` parameters in the `[Maild]` [section](#) of **Dr.Web MailD** configuration file) and the `stat` parameter value is `yes`.

Example:

```
InPoolOptions = auto, stat = yes
```

Note that for **Sender** and **Receiver** components statistics is collected unconditionally because additional parameters (such as `timeout`, `stat` and others) are not specified for their pools and even if so, they have no effect.

Aggregated statistics is saved to the log file as a message of `Debug` verbosity level upon receipt of `SIGUSR1` signal or **Dr.Web MailD** shutdown. Note that if the set log verbosity level is less detailed than `Debug`, statistics is not logged.

Statistics record looks as in the following example:

```
size = 50, active = 0, pending = 0, min = 50, max = 500, threshold = 50
```



where:

- `size` – current size of the pool (number of threads in the pool);
- `active` – number of threads in active state at the moment of record creation;
- `pending` – number of tasks that are wait for free thread in the pool at the moment of record creation;
- `min` – minimal possible number of threads in the pool;
- `max` – maximal possible number of threads in the pool;
- `threshold` – step by which the number of threads in the pool is increased/decreased if necessary.

Statistics is not aggregated. Upon receipt of time signal, the current state of the pool is fixed.

Moreover, separate files with statistics are created for certain pools. Statistics is also saved to such files upon receipt of `SIGUSR1` signal and **Dr.Web MailD** shutdown.

Files with statistics are named according to the following patterns:

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

where:

- `name` – name of the component (name of the [corresponding module](#) without "drweb-" part).
- `callback` – callback of the **Receiver** interface.
- `cli` – for `client` connections.
- `srv` – for `server` connections.
- `unique-id` – for modules started with the [unique identifier](#).
- `thr` – for a [thread pool](#).

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

Each record starts with the following:

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

where:

- `start` – time of the previous statistics record,
- `curr` – time of the current statistics record;
- `period` – time interval between the previous and current records.

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

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

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

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



thr output looks as follows:

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

where:

- first line contains information on the maximum and minimum number of threads in one 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 additional 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.

RFC Standards

Dr.Web MailD operates according to requirements of the following [RFC documents](#):

Number	Name
1123	Requirements for Internet Hosts - Application and Support
1652	SMTP Service Extension for 8bit-MIME transport
1830	SMTP Service Extensions for Transmission of Large and Binary MIME Messages
1870	SMTP Service Extension for Message Size Declaration
1894	An Extensible Message Format for Delivery Status Notifications
2033	Local Mail Transfer Protocol
2034	SMTP Service Extension for Returning Enhanced Error Codes
2045 – 2049	Multipurpose Internet Mail Extensions
2222	Simple Authentication and Security Layer (SASL)
2231	MIME Value and Encoded Word Extensions
2245	Anonymous SASL Mechanism
2289	A One-Time Password System
2444	The One-Time-Password SASL Mechanism
2505	Anti-Spam Recommendations for SMTP MTAs
2646	The Text/Plain Format Parameter
2821	Simple Mail Transfer Protocol
2822	Internet Message Format
2831	Using Digest Authentication as a SASL Mechanism
2945	The SRP Authentication and Key Exchange System
3174	US Secure Hash Algorithm 1 (SHA1)



Adjustment and Startup

Dr.Web for UNIX mail servers can be started with default settings, but if you want to ensure optimal performance, you may adjust it according to your specific requirements.

All **Dr.Web for UNIX mail servers** settings are stored in the following three configuration files located in the `%etc_dir` directory:

- `maild_<MTA>.conf` – file with **Dr.Web MailD** general settings;
- `agent.conf` – file with **Dr.Web Agent** settings;
- `monitor.conf` – file with **Dr.Web Monitor** settings.

If all files of **Dr.Web for UNIX mail servers** are located in their default directories, basic setup of the solution can be performed with the `configure.pl` script. The default directory of the script is `%bin_dir/maild/scripts/`.

After startup, the script prompts you to specify values for some essential parameters and writes them to the `maild_<MTA>.conf` configuration file.

Also you should use the `configure_mta.sh` script. This script is responsible for setting up interaction between **Dr.Web for UNIX mail servers** and the currently used mail system. After startup, the script checks whether the required mail system is installed. If it appears to be missing, the script finishes its operation. If the required mail system is installed, the script asks the user several questions on essential settings for basic setup and makes the necessary changes in corresponding configuration files.

Other parameters that are necessary for interaction with mail system must be adjusted manually, by editing the `maild_<MTA>.conf` configuration file of **Dr.Web MailD**.



Part of the `<MTA>` file name depends on the name of the MTA integrated with **Dr.Web for UNIX mail servers**.

Dr.Web MailD Configuration Files

Main configuration file

Dr.Web MailD settings (including interaction between MTA and mail systems as well as usage of plug-ins for mail check) are specified in the `%etc_dir/maild_<MTA>.conf` configuration file.

For description of the configuration file structure and parameter types, see [Configuration Files](#).

For description of the special parameter types used in **Dr.Web MailD** settings, see [Special parameter types](#).

For description of the `Lookup`, `LookupLite` and `Storage` special types, see [Lookup](#).

For the list of configuration file sections, see [Sections of main configuration file](#).



Part of the `<MTA>` file name depends on the name of the MTA integrated with **Dr.Web for UNIX mail servers**.

Configuration files of plug-ins

Each [plug-in](#) uses its own configuration file. Configuration files of plug-ins are located in the same directory as the main **Dr.Web MailD** configuration file: `%etc_dir`. Configuration files of plug-ins are named according to the following pattern `plugin<name>.conf`, where `<name>` – name of the plug-in. For example, configuration file of **Drweb plug-in** has the following name: `plugin_drweb.conf`.



If required, you can configure each plug-in to use configuration files which names do not match the pattern. To do that, make the corresponding adjustments in the [Filters] [section](#) of the main **Dr.Web MailD** configuration file.

Special Parameter Types

This chapter provides you with detailed description of the following special parameter types:

- **TLS/SSL settings (TLSSettings)** — settings for connection secured using the TLS and SSL protocols.
- **Thread pool settings (pool options)** — settings of a thread pool.

For description of the **Lookup**, **LookupLite** and **Storage** special parameters, see [Lookup](#).

TLS/SSL settings (TLSSettings)

Settings for connection secured using the TLS and SSL cryptographic protocols.

Settings are defined as a sequence of `PARAMETER VALUE` pairs, separated by commas. File path specified as `VALUE` is case-sensitive (because it is a convention for UNIX-like systems).

The current version supports the following settings:

- `use_sslv2` {yes | no} - enable or disable use of the **SSLv2** protocol. By default the use of **SSLv2** is disabled, because this protocol is insecure.
- `use_sslv3` {yes | no} - enable or disable use of the **SSLv3** protocol. **SSLv3** is enabled by default.
- `use_tlsv1` {yes | no} - enable or disable use of the **TLSv1** protocol. **TLSv1** is enabled by default.
- `private_key_file` {path to the file} - absolute path to a private key file. The key must be specified in the `PEM` format. Key encryption is supported. The parameter is required for server configuration. Default parameter value is not set.
- `private_key_password` {string} - password for the key specified in the `private_key_file` parameter. Default parameter value is not set.
- `certificate` {path to the file} - path to a certificate file with a signed public key. The value of this parameter must be specified together with the value of the `private_key_file` parameter. This parameter is required for server configuration. Default parameter value is not set.
- `verify_mode` {none | peer | client_once | fail_if_no_peer_cert} - sets peer certificate verification mode:
 - `none` - skip peer certificate verification. This value is set by default;
 - `peer` - verify a peer's certificate. The parameter is ignored in the client mode if the server does not send a certificate for anonymous encryption. This value is set by default for client connections;
 - `client_once` - request a certificate only when connection is established for the first time (disable certificate request during TLS handshake if connection is already established). The value can be used only together with the `peer` value.
 - `fail_if_no_peer_cert` - enable the server to treat missing of a client's certificate as an error. The value can be used only together with the `peer` value.

Examples:

```
verify_mode peer, verify_mode client_once
verify_mode none
```

If `peer` and `none` values are specified together, the last specified value is used.



- o `verify_ca` {the path to the file | the path to the directory} - the absolute path to a file or directory where **CA** certificates in the `PEM` format are located. These certificates are used to validate a peer's certificate.
- o `cipher_list` {string} - a list of allowed encryption algorithms. Use the `man ciphers` command to get information on the format of encryption algorithm list (**OpenSSL** must be installed).

Thread pool options

The options have combined value. Parameters are separated by commas.

At first, number of threads in a pool is defined:

- o `auto` - number of threads in a pool is automatically detected, depending on the current system load;
- o `N` - non-negative integer. Only `N` threads in a pool are active and new threads cannot be created;
- o `N-M` - positive integers, $M \geq N$. At least `N` threads in a pool are active, and new threads can be created as required until their number reaches `M` value (if it is necessary to set a constant number of threads in the pool, set $M=N$).

After that, the following additional parameters can be specified:

- o `timeout = {time}` - if a thread is not active during the specified time period, it is closed. This parameter does not affect the first `N` threads which wait for requests indefinitely. Default value: `2m`
- o `stat = {yes|no}` - statistics on threads in a pool. It is saved each time `SIGUSR1` system signal is received or upon **Dr.Web MailD** shutdown, to the directory specified in the value of the `BaseDir` parameter from the [General] [section](#). Default value: `no`
- o `log_level = {Quiet|Error|Alert|Info|Debug}` - log [verbosity level](#) for threads in a pool. If the value is not explicitly specified, value of the `LogLevel1` parameter from the [Logging] [section](#) is used.
- o `stop_timeout = {time}` - maximum time for a working thread to stop (for example, when program finishes its operation or when it is necessary to decrease the number of threads in a pool).

Example:

```
InPoolOptions = auto
```

Number of threads is detected automatically, [internal statistics](#) is not collected (except for **Receiver** and **Sender**, as statistics is collected permanently for these components).

Example: (for **Notifier**)

```
PoolOptions = 25, stat=yes
```

In the example, number of threads in the pool always equals 25, [internal statistics](#) is collected.



For **Sender** and **Receiver** components `auto` value equals to 2-500, and for other **Dr.Web MailD** components the value equals to 2-1000. Be careful when changing a number of threads in pools. For details, refer to [Optimizing operation and use of system resources](#).

For **Sender** and **Receiver** components additional parameters of the thread pool (such as `timeout`, `stat` and others) are not specified and even if the values are set, they take no effect as **Sender** and **Receiver** permanently [collect statistics](#) on the thread pools.



Lookup

`Lookup` is generalized interface for searching objects and receiving their values. Values are separated by commas. Some values have a prefix denoting a specific type of Lookup. The prefix is separated by a colon:

```
[prefix1:]value1, [prefix2:]value2, ...
```

If a prefix is not specified, values are used directly.

Special symbols

You can use the following *special symbols* in Lookup queries:

- `$s` - is substituted with the requested element. For example, if an address is requested, `$s` is substituted with the full address (without angle brackets), if a domain is requested, `$s` is substituted with the domain name.
- `$d` - if an address is requested, `$d` is substituted with the domain part of this address. Otherwise, the full address is inserted.
- `$u` - if an address is requested, `$u` is substituted with the username from this address. If a domain is requested, an empty string is inserted.
- `$$` - is substituted with a single `$`.



Note that in some cases values of special symbols cannot be determined for substitution during `Lookup`. Primarily it concerns parameters used by **Receiver component** to **check SMTP restrictions** (SMTP/LMTP on the INTRO stage of SMTP-session (connection of a client)). On this stage:

- For all checks whether the client's IP address is in the network list (see the **ProtectedNetworks** parameter in the [Maild] [section](#)):
 - `$s` - IP address of the connected client.
 - `$d`, `$u` - empty as the corresponding value is not determined.
- For all checks whether the client's domain is in the domain list (see the **ProtectedDomains** parameter in the [Maild] [section](#)):
 - `$s` - domain name of the client's host (if FQDN was resolved), otherwise - its IP address.
 - `$d` = `$s`.
 - `$u` - empty as the corresponding value is not determined.

On next SMTP session stages (MAIL FROM, RCPT TO) values of all special symbols are determined:

- `$s` - full address of user@domain.
- `$d` - domain (domain part of the address).
- `$u` - user name (user part of the address).

Prefixes

You can use the following prefixes to specify a data source:

- `value` - specify the requested value after this prefix. This is default prefix and can be skipped. You can use this prefix when, for example, the value contains ':' symbol.
- `file` - the value is a file path. Each value in the file must be set in a separate line. That allows rapid searching, because assortment and binary search can be used.



Please note that for parameters having special restricted type **LookupLite**, allowed only these two prefixes, and other prefixes, presented below, are prohibited!

- `regex` - the value is a regular expression (compatible with **Perl** regular expressions) - the object is searched by a substring, absolute matching is not required.
- `rfile` - the value is a file path. The file contains a set of regular expressions (compatible with



Perl regular expressions), and each of them must be set in a separate line. The object is searched by a substring, absolute matching is not required.



Please note that contents of files used as data source of **file** and **rfile** types are not checked in advance. Before using the files, ensure that:

- files are of the text type (contain only text lines);
- files do not contain empty and "garbage" lines (such as separators or comments) which cannot be used as **Lookup** values;
- regular expressions (for **rfile**) are given correctly to match the **Perl** regular expressions syntax;
- files do not contain superfluous data;
- file size is not too large, as when a file of large size is read, a memory allocation error can occur (in this case, **Dr.Web MailD** is terminated).

- **ldap** – the value is the path to the **LDAP** server. The value is set in the following format:

```
[param1=val1|param2=val2|...|] ldap_url
```

where **ldap_url** – is an URL of **LDAP** query. Specify **param1**, **param2** and others to override their values in the **Lookup** query (the values are set in the **[LDAP]** section of **Dr.Web MailD** configuration file). You can specify only those parameters, for which this option is explicitly stated (see description of the section parameters). For parameters that are not specified, values from the **[LDAP]** section of **Dr.Web MailD** configuration file are used.

URL of **LDAP** query (**ldap_url**) is as follows:

```
ldap://hostport/dn[?attrs[?scope[?filter[?exts]]]]
```

where:

- **hostport** – host name (can be specified with a port number separated by a colon - ":portnumber");
- **dn** – database name where search is performed;
- **attrs** – comma separated list of request attributes;
- **scope** – can have one of the following values: **base**, **one**, **sub**;
- **filter** – filter name;
- **exts** – set of **LDAP** and/or **API** extensions.

Example:

```
ldap://ldap.example.net/dc=example,dc=net?cn,sn?sub?(cn=*)
```

- **odbc**, **postgres**, **oracle**, **mysql**, **firebird**, **sqlite** – the value is an SQL-query to the database in the corresponding **DBMS** (for **ODBC** – to **DSN** data source). SQL-query is of the following format:

```
[param1=val1|param2=val2|...|] sql_request
```

where **sql_request** – is a text of SQL query to the database and **param1**, **param2** and others are parameters from the **[<DATASOURCE>]** section of **Dr.Web MailD configuration file** for the given **Lookup** (**<DATASOURCE>** – is name of used **DBMS** or **ODBC**, and is always equal to prefix which is used in the given **Lookup**: that is, **[ODBC]**, **[PostgreSQL]**, **[Oracle]**, **[MySQL]**, **[Firebird]**, **[SQLite]**). From this section, only those parameters can be specified, for which this option is explicitly stated (see description of the section parameters). Pairs **param=value** must be included only if it is necessary to override their values in the given **Lookup** query. For parameters that are not specified, values from the **[<DATASOURCE>]** section of **Dr.Web MailD** configuration file are used.



You can use *special symbols* as in the Lookup queries.

**Important notes:**

1. If a `SELECT` query returns records which contain more than one string (query of the following type: `SELECT field1,field2,... FROM ...`), all strings returned by DBMS are assigned to Lookup values. It is not recommended to use such queries because different DBMS can return such responses in different formats (some DBMS separate fields with a whitespace, some – with a comma or semicolon).
2. Lookups with the `sqlite` prefix are used for interaction with **SQLite** version 3.x. See also [notes on features of working with SQLite](#) DBMS

- `cdb` – the value is an alphanumeric name of the key in the **CDB** database. **CDB** database does not support the SQL query input language, that is why driver emulates the single SQL command for operating with lookups:

```
select * from @tablename where key='@string'
```

where `@tablename` must be changed to the name of any file specified as a source item in the [CDB] [section](#) of **Dr.Web MailD** configuration file.

You can use *special symbols* in Lookup queries.

Example:

```
cdb:skipdomains=regex:^inbox|select * from my_file where key='$s'
```

Note that the `skipDomains` parameter, locally overridden in this `Lookup`, is of the [LookupLite](#) type (that is, `Lookup` for which only `file:` and `value:` prefixes are allowed).

- `berkeley` – enables interaction with **Berkeley DB**. Format of the query is similar to that of `cdb` prefix. Parameters from the [Berkeley] [section](#) of **Dr.Web MailD configuration file** are used.

You can use *special symbols* as in the Lookup queries.

Local overriding of parameters in Lookup queries

After a prefix, you can optionally specify a list of values for the `SkipDomains` and `OnError` parameters to be used in this `Lookup`. The local values are specified in the following format:

```
NAME1 = VALUE1 | NAME2 = VALUE2 | ... |
```

where:

- `NAME` – case-insensitive parameter name;
- `VALUE` – parameter value.

If a parameter is not locally overridden in a `Lookup`, their default values or values from the corresponding [`<DATASOURCE>`] [section](#) of **Dr.Web MailD configuration** are used (where `<DATASOURCE>` corresponds to the `Lookup` prefix).



Note that usage of `SkipDomains` in `Lookup` has no meaning if the `Lookup` is to determine parameter values on those SMTP session stages when the domain cannot be determined (that is, `$d` special symbol is empty). Domain cannot be determined on the `INTRO` stage of SMTP session (see above).

Features of Lookups processing

Please note that during `Lookup` processing, **Dr.Web MailD** waits for connection to a data source (DBMS or LDAP server) during a timeout specified in the the data source settings or overridden locally in the `Lookup` prefix. This can slow down the performance of **Dr.Web MailD** if the network connection is not stable or specified connection parameters are incorrect.

If connection attempt fails during the timeout, the error is fixed and processed according to the `OnError` parameter value specified either in the data source settings or overridden locally in the



Lookup prefix.

If a `Lookup` is used as the `Router` parameter value (specified in the [Sender] [section](#)) and the `OnError=exception` error handling mode is used (this mode can be set in the data source settings or locally overridden in the `Lookup` value expression), the state when `Sender` cannot receive required route from the used data source is handled as an error in `Sender` operation. Error message is logged. In this case:

- in the [synchronous](#) mode, `Receiver` returns the SMTP error code `451` (Requested action aborted: local error in processing) to the message sender, and the email message is deleted from all queues of **Dr.Web MailD**.
- in the [asynchronous](#) mode, the message is marked as 'stalled' and `Sender` tries to send it at intervals specified in the `StalledProcessingInterval` parameter value.



Incorrect Lookup string can cause **Dr.Web MailD** to terminate on its startup (when reading a configuration file) if **Dr.Web MailD** cannot parse the Lookup string structure and recognize the type.

It is recommended to use a special [utility](#) for testing correctness of Lookups.

Note that full information on queries to data sources sent via Lookups is output only if the [log verbosity level](#) is not less detailed than `DEBUG`.

Lookup Usage Examples

Example 1:

```
ProtectedDomains = "odbc:select domain from maild where domain='\$s'"
```

With this query, all messages from the domain found in `domain` column of `maild` table in ODBC data source are marked as messages belonged to the protected domain.

Example 2:

```
ProtectedEmails = file:%etc_dir/email.ini, localhost,  
ldap:skipdomains=file:/home/trusted_domains|ldaps:///??sub?(mail=$s)
```

With this query, the following addresses are marked as protected:

- all addresses from the `%etc_dir/email.ini` file;
- `localhost` address;
- all addresses found with the `ldap:///??sub?(mail=$s)` LDAP query except for those that are listed at `/home/trusted_domains` (these addresses are not queried).

Example 3:

```
Router = mysql:select routerinfo from maild where email='\$s', foo  
inet:234@foo.ru
```

With this query, it is checked whether the certain address is listed in the `email` column of the `maild` table in **MySQL** database. If the address is in the database, the message is dispatched to the address found in `routerinfo` column, otherwise it is sent to the `inet:234@foo.ru` address for all recipients with `foo` in the address.

Lookups can be also used in [Rules](#).

Example 4:

```
"rcpt:ldap:///?rules?sub?(mail=$s)" cont
```

This query allows receiving the `rules` field for all mail LDAP fields that contain the receiver address. The `rules` field contains settings to be applied to this recipient.

Please note that all `CONDITIONS` must be enclosed in quotation marks, as a `CONDITION` part can contain special symbols (for example, round brackets).



Thus, if you write the following:

```
rcpt:"ldap:///?rules?sub?(mail=$s)" cont
```

an error will occur :

```
Mon Jun 29 18:53:01 2009 [3081262768] maild.rules ERROR '(' can not follow  
'"ldap:///?rules?sub?'  
Mon Jun 29 18:53:01 2009 [3081262768] maild.rules ERROR error in parse  
condition:  
'rcpt:"ldap:///?rules?sub?(mail=$s)" cont'
```

Example 5:

```
"any:sqlite:select skipaddr from domain where skipaddr = '$s'" cont  
scan=all:-drweb
```

This query enables checking of addresses. If sender's or recipient's addresses are in the `skipaddr` field of `domain` table of **SQLite** database, **Drweb plug-in** is not used for them.

Lookup Usage Restrictions. LookupLite Type

Sometimes it is impossible to use Lookup of [certain types](#) with the full set of prefixes (not all prefixes are allowed). In this case, a special Lookup type – `LookupLite` is used.

`LookupLite` is a [value type](#) similar to `Lookup`, but you can specify only the following types of the `Lookup`:

- o **value**: (this is an optional prefix and can be absent for single value)
- o **file**:

`LookupLite` is used in:

- settings of `Lookups` (e.g. **SkipDomains** parameter for each `Lookup`);
- settings of [plug-ins](#).

At any attempt to specify a forbidden type of `Lookup`, the following message is output to the log:

```
Wed Jun 10 14:02:20 2009 [4160149200] Modifier ERROR Error in init lookup  
[cdb:select * from /root/mail/base_file_for_CDB.txt where key='domain']:  
can't use this lookup here.
```



It is recommended to use a special [utility](#) for testing correctness of `Lookups` and `LookupLite`.

Storage Data Type

`Storage` data type describes objects used to store data. Syntax of the `Storage` type is similar to [Lookup](#) except for the following differences:

- another set of prefixes is used;
- the `$s` special symbol cannot be used.

You can use the following prefixes:

- o **value** — as well as for `Lookups`, the value is directly specified after this prefix. For example, you can use this prefix, when the value contains `'` symbol.



- o `odbc`, `oracle` — these prefixes are similar to those for Lookups. In SQL expressions, it is possible to specify values to be saved in the following format:

```
:name<type>
```

where `name` — name of the saved object (each parameter has its own list of possible names),
`type` — data type which is used for saving the parameter value in the storage.

- o `postgres`, `mysql`, `sqlite`, `firebird` — syntax is similar to the previous one with the following difference: data type `char(<length>)` cannot be used, for string data it is necessary to use `varchar_long` SQL data type.

For interaction with databases in the corresponding DBMS, the `odbc`, `oracle`, `postgres`, `mysql`, `sqlite`, `firebird` prefixes are used with the parameters from the corresponding [`DATASOURCE`] sections of the [Dr.Web MailD configuration file](#).

Example:

```
"odbc:insert into plugin_stat values  
(:plugin_name<varchar_long>, :size<int>, :num<int>);"
```

Please note that quotation marks are required in this example because the query text contains commas.

Sections of Main Configuration File

Dr.Web MailD configuration file, as any other configuration file of **Dr.Web for UNIX mail servers** components, is textual and consists of sections (see general description of [Dr.Web for UNIX mail servers configuration files](#)).

Dr.Web MailD configuration file consists of the following sections:

Sections with main parameters of **Dr.Web MailD** operation:

- [\[General\]](#) — contains general settings of **Dr.Web MailD**. This section is mandatory.
- [\[Maild\]](#) — contains general settings of **MailD core**. This section is mandatory.
- [\[MailBase\]](#) — contains settings of internal database of email messages included into **MailBase**. This section is mandatory;
- [\[Notifier\]](#) — contains settings of **Notifier** (used for sending notifications, reports and DSN). This section is mandatory.
- [\[Quarantine\]](#) — contains settings of **Quarantine**. This section is mandatory.
- [\[Filters\]](#) — contains settings of all used plug-ins and their launch order during processing of an email message. This section is mandatory.
- [\[Rule\]](#) — contains default settings for parameters used in [Message processing Rules](#) (for initialization of parameter values if they are not initialized by any rule). This section is mandatory and must be specified before the `[Rules]` section.
- [\[Rules\]](#) — contains rules for mail processing management. This section is optional. Can be empty or absent if processing rules are not used or they are moved to the [internal database](#).
- [\[Stat\]](#) — contains settings of statistics collection. This section is optional and can be absent.
- [\[Reports\]](#) — contains settings of reports generating. This section is optional and can be absent.
- [\[Logging\]](#) — contains logging settings. This section is optional and can be absent.

The following sections contain settings of SASL authentication if it is used. If SASL authentication is not used, both sections can be absent. If SASL is used, both sections are mandatory:

- [\[SASL\]](#) — contains settings of SASL authentication.
- [\[Cyrus-SASL\]](#) — contains settings of the `cyrus-sasl` SASL driver (in the current version, only this



driver can be used).

The following sections contain parameters of interaction with different mail systems (MTA):

- **[Receiver]** – contains settings for **Receiver** which operates directly via the SMTP/LMTP protocol and also interacts with the **Exim**, **Zmailer** and **Postfix** (if **Postfix** does not use the **Milter** protocol) mail systems. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** operates in **SMTP/LMTP proxy** mode or interacts with **Exim**, **Zmailer** or **Postfix** (if **Postfix** does not use the **Milter** protocol).
- **[Sender]** – contains settings for **Sender** which operates directly via the SMTP/LMTP protocol and also interacts with all mail systems except **CommuniGate Pro**. This section can be absent if **Dr.Web for UNIX mail servers** interacts only with **CommuniGate Pro**.

Please note that both sections, **[Receiver]** and **[Sender]**, are mandatory if **Dr.Web for UNIX mail servers** operates in **SMTP/LMTP proxy** mode or interacts with **Exim**, **Zmailer** or **Postfix** (if **Postfix** does not use **Milter** protocol). Only the **[Sender]** section is mandatory if **Dr.Web for UNIX mail servers** interacts with **Qmail**, **Courier** or other mail system which uses **Milter** protocol (for example, **Sendmail** or **Postfix**).

- **[Courier]** – contains settings of interaction with **Courier** mail system. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** is integrated with **Courier** mail system.
- **[CgpReceiver]** – contains settings of **Receiver** which interacts with the **CommuniGate Pro** mail system. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** is integrated with the **CommuniGate Pro** mail system.
- **[CgpSender]** – contains settings of **Sender** which interacts with the **CommuniGate Pro** mail system. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** is integrated with the **CommuniGate Pro** mail system.

Please note that both **[CgpReceiver]** and **[CgpSender]** sections are mandatory if **Dr.Web for UNIX mail servers** interacts with the **CommuniGate Pro** mail system (otherwise, both sections can be absent).

- **[Milter]** – contains settings of interaction with mail systems which use the **Milter** protocol. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** is integrated with any mail system which uses **Milter** protocol (for example, **Sendmail** or **Postfix**).
- **[Qmail]** – contains settings of interaction with the **Qmail** mail system. This section is optional and can be absent. However, this section is mandatory if **Dr.Web for UNIX mail servers** is integrated with the **Qmail** mail system.
- **[IMAP]** – contains settings of **Dr.Web for UNIX mail servers** operation as a proxy between mail system and mail clients that use the **IMAP** protocol. This section is optional and can be absent if **Dr.Web for UNIX mail servers** is not used as a proxy between mail system and mail clients that use the **IMAP** protocol.
- **[POP3]** – contains settings of **Dr.Web for UNIX mail servers** operation as a proxy between mail system and mail clients that use the **POP3** protocol. This section is optional and can be absent if **Dr.Web for UNIX mail servers** is not used as a proxy between mail system and mail clients that use the **POP3** protocol.

In the configuration file, only sections that contain parameters of interaction between **Dr.Web for UNIX mail servers** and integrated MTA are mandatory. Sections with parameters of interaction with unused MTA are ignored, even if the sections are present. For details on using configuration file sections on integration with different mail systems, see [MTA connections](#).

The following sections contain parameters of connection to different data sources (LDAP, relational DBs, text files):

- **[LDAP]** – contains parameters of connections to of connection to LDAP data sources.



- [\[Oracle\]](#) – contains parameters of connections to of connection to **Oracle** DBMS;
- [\[ODBC\]](#) – contains parameters of connections to of connection to data sources via ODBC (if the corresponding ODBC driver is present).
- [\[SQLite\]](#) – contains parameters of connections to of connection to **SQLite** DBMS.
- [\[Firebird\]](#) – contains parameters of connections to of connection to **Firebird** DBMS.
- [\[PostgreSQL\]](#) – contains parameters of connections to of connection to **PostgreSQL** DBMS.
- [\[MySQL\]](#) – contains parameters of connections tottings of connection to **MySQL** DBMS.
- [\[CDB\]](#) – contains parameters of connections to of connection to the **CDB** textual database.
- [\[Berkeley\]](#) – contains parameters of connections to of connection to the **Berkeley** textual database.

Please note that these sections contain only common parameters used in [Lookup](#) and [Storage](#) by default. Some parameters can be locally overridden in a `Lookup` and `Storage` (if that is marked in their descriptions).

It is recommended to adjust each connection to a data source via native connection (if possible), as connection via **ODBC** works more slowly because of its universality.

In the configuration file, any sections with parameters of connection to data sources can be present. Only those sections are really used, that correspond to prefixes specified in `Lookups` and `Storages`. If parameters are not used, they cannot influence **Dr.Web MailD** operation, even if the section is present in the file.

Sections with proxy parameters:

- [\[ProxyClient\]](#) – contains settings of an internal proxy client that provides interaction of **Receiver** and **Sender** with the **MailD core** main module).
- [\[ProxyServer\]](#) – contains settings of an internal proxy server (**MailD core** module uses it for processing queries from internal proxy clients).



Please note that if a section is not present in the configuration file, parameters from these section have default values. These default values are specified in this manual in the description of each parameter.

It is **strongly not recommended** to add sections and parameters to the configuration file which was automatically generated upon product installation if the sections and parameters are described in the current document but absent in the file. This is due to the fact that the parameters are specific for a certain mail system and adjustment of their values after integration with another file system can prevent **Dr.Web for UNIX mail servers** from working correctly.

General Parameters

This chapter provides you with description of the configuration file sections which contain general parameters of **Dr.Web for UNIX mail servers** and its main **Dr.Web MailD** components.

Generally, all these sections are always present in the configuration file.

[General] Section

In the `[General]` section, general settings of **Dr.Web MailD** are specified:

```
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](#). Restart of `%MAILD%>` is required.

Default value:

```
BaseDir = %var_dir/
```



<code>MaxTimeoutForThreadActivity = {time}</code>	<p>Maximum time for a thread to close.</p> <p>This parameter is used on system restart or shutdown. Total time for the system to shut down is calculated in the following way: number of pools and the MaxTimeoutForThreadActivity parameter value are multiplied together, and then a certain time constant is added to the result.</p> <p><u>Default value:</u> MaxTimeoutForThreadActivity = 30s</p>
<code>IpTimeout = {time}</code>	<p>Timeout for establishing connection between components or waiting response.</p> <p>After timeout is expired the connection is disconnected. Also an error of interaction between components is fixed. This error is processed as specified in the ProcessingErrors parameter value (in the [Maild] section).</p> <p>In the current version, value of this parameter cannot be changed when reloading the system by SUGHUP signal.</p> <p>Timeout period must be approximately equal to average time of processing an email message by plug-ins. It is particularly significant if the before-queue mode is enabled (message is checked by plug-ins from the list specified in the BeforeQueueFilters parameter value in the [Filters] section).</p> <p><u>Default value:</u> IpTimeout = 40s</p>
<code>Hostname = {string}</code>	<p>Name of the host used by Dr.Web for UNIX mail servers.</p> <p><u>Default value:</u> Hostname =</p>

[Maild] Section

In the [Maild] section, general setting for **Dr.Web MailD** proper operation are specified:

<code>ProtectedNetworks = {Lookup}</code>	<p>List of networks protected by Dr.Web MailD. Values are specified in the CIDR format.</p> <p>This parameter is used to specify trusted networks in the corresponding Vaderetro plug-in parameters and if <code>trust_protected_networks</code> is specified in the SessionRestrictions parameter in the [Receiver] section.</p> <p>Please note that the parameter value is Lookup.</p> <p>Using of Lookup has series of restrictions, described below.</p> <p>Example: ProtectedNetworks = 10.0.0.0/24, 127.0.0.0/8, "mysql:select net from networks where net='\$s'"</p> <p><u>Default value:</u> ProtectedNetworks = 127.0.0.0/8</p>
<code>ProtectedDomains = {Lookup}</code>	<p>List of domains protected by Dr.Web MailD.</p> <p>This parameter is used to specify trusted domains if <code>trust_protected_domains</code> is specified in the SessionRestrictions parameter in the [Receiver] section.</p> <p>Please note that the parameter value is Lookup.</p>



	Example: <code>ProtectedDomains = example.ru, example.com</code>
	Default value: <code>ProtectedDomains =</code>
<code>IncludeSubdomains = {logical}</code>	Include subdomains in the list of protected domains.
	Default value: <code>IncludeSubdomains = yes</code>
<code>InPoolOptions = {pool options}</code>	Settings for a pool of threads that process messages before they are queued.
	Default value: <code>InPoolOptions = auto</code>
<code>OutPoolOptions = {pool options}</code>	Settings for a pool of threads that process messages after they are queued.
	Default value: <code>OutPoolOptions = auto</code>
<code>RedirectMail = {email address}</code>	R email address where messages are sent to, when Redirect action is used (if the address is not specified as a parameter of the Redirect action).
	Default value: <code>RedirectMail = root@localhost</code>
<code>OnlyTrustedControlMails = {logical}</code>	Send control messages only from the protected network. If Receiver did not provide information about client's IP address, set <code>GetIpFromReceivedHeader = Yes</code> , that enables MTA to add the correct <code>Received</code> header to all messages before transmitting them to Dr.Web for UNIX mail servers . To ensure correct work of control messages, all outgoing email traffic of clients must be scanned by Dr.Web for UNIX mail servers .
	Default value: <code>OnlyTrustedControlMails = Yes</code>
<code>MaxScore = {numerical value}</code>	R Maximum message score. If message score exceeds this parameter value, actions specified in the <code>MaxScoreAction</code> parameter are applied to this message and message check stops. This parameter is checked before a message is transmitted to plug-ins and after the message is checked by each plug-in.
	Default value: <code>MaxScore = 10000</code>
<code>MaxScoreAction = {actions}</code>	R Actions applied to the message when its score exceeds the threshold value specified in the <code>MaxScore</code> parameter. More than one action can be specified in this parameter (first action is mandatory, others are optional). Mandatory action can be one of the following: pass, discard, reject, tempfail.



	<p>Additional actions can be the following: quarantine, redirect, add-header, score.</p> <p>If <code>reject</code> action is specified and value of the <code>UseCustomReply</code> parameter is set to <code>yes</code>, the SMTP response is taken from the <code>ReplyMaxScore</code> parameter (see below). After all actions are applied, message check is considered finished.</p> <p><u>Default value:</u> MaxScoreAction = <code>reject</code></p>
<p>MaxMimeParts = {numerical value}</p>	<p>Maximum number of MIME parts in a message.</p> <p>If the value is set to 0, check is not performed. If the number of MIME parts in a message exceeds the specified threshold value, its processing is aborted and actions specified in the <code>ProcessingError</code> parameter are applied to the message (see below).</p> <p><u>Default value:</u> MaxMimeParts = 1000</p>
<p>MaxNestedMimeParts = {numerical value}</p>	<p>Maximum number of nested MIME parts in the message.</p> <p>If the value is set to 0, check is not performed. If the number of nested MIME parts in a message exceeds the specified threshold value, its processing is aborted and actions specified in the <code>ProcessingError</code> parameter are applied to it (see below).</p> <p><u>Default value:</u> MaxNestedMimeParts = 100</p>
<p>LicenseLimit = {actions}</p>	<p>R Actions applied to messages that were not scanned because of license limitations.</p> <p>More than one action can be specified in this parameter (first action is mandatory, others are optional).</p> <p>Mandatory action can be one of the following: pass, discard, reject, tempfail.</p> <p>Additional actions can be the following: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> LicenseLimit = <code>pass</code></p>
<p>EmptyFrom = {actions}</p>	<p>R Actions applied to messages that have an empty <code>From</code> header.</p> <p>An empty <code>From</code> header is typical when sending DSN (they must have an empty FROM header to meet the protocol requirements); spammers also often leave this header empty.</p> <p>More than one action can be specified in this parameter (first action is mandatory, others are optional).</p> <p>Mandatory action can be one of the following: continue, discard and reject.</p> <p>Additional actions can be the following: quarantine, redirect, add-header, score.</p> <p><u>Default value:</u> EmptyFrom = <code>continue</code></p>
	<p>R Action applied to messages which invoked errors during scanning.</p>



ProcessingErrors = {actions}	<p>More than one action can be specified in this parameter (first action is mandatory, others are optional).</p> <p>Mandatory action can be one of the following: pass, discard, reject, tempfail.</p> <p>Additional actions can be the following: quarantine, redirect, notify, add-header, score.</p> <p>Please pay attention to the features of errors handling, presented below.</p> <p><u>Default value:</u> ProcessingErrors = pass</p>
RulesLogLevel = {log level}	<p>Log verbosity level for Rule processor</p> <p>The following levels are allowed:</p> <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug <p><u>Default value:</u> RulesLogLevel = Alert</p>
PidFile = {path to file}	<p>Path to the PID file of drweb-maild process.</p> <p><u>Default value:</u> PidFile = %var_dir/run/drweb-maild.pid</p>

When message is blocked by any **Dr.Web for UNIX mail servers** component (i.e. **reject action** performed), 550 5.7.0 error code and a text message is used for SMTP-reply. Text for the message can be specified in parameters described below.

UseCustomReply = {logical}	<p>R Enables usage of custom messages in SMTP sessions.</p> <p>These messages are sent as an SMTP reply when incoming message is rejected.</p> <p><u>Default value:</u> UseCustomReply = No</p>
ReplyEmptyFrom = {string}	<p>R Reply that is sent when EmptyFrom action is applied and if:</p> <ul style="list-style-type: none">• EmptyFrom = reject;• UseCustomReply = Yes. <p>You can specify only text part of the reply: "550 5.7.0 <Text>".</p> <p>Text must be enclosed in quotation marks if it contains white spaces.</p> <p><u>Default value:</u> ReplyEmptyFrom = "DrWEB maild: Messages from <> are blocked by administrator."</p>
ReplyProcessingError = {string}	<p>R Reply that is sent when ProcessingError action is applied and if:</p> <ul style="list-style-type: none">• ProcessingError = reject;• UseCustomReply = Yes.



	<p>You can specify only text part of the reply: "550 5.7.0 <Text>".</p> <p>Text must be enclosed in quotation marks if it contains white spaces.</p> <p><u>Default value:</u> ReplyProcessingError = "DrWEB maild: Message is rejected due to software error."</p>
<pre>ReplyMaxScore = {string}</pre>	<p>R Reply that is sent when MaxScoreAction action is applied and if:</p> <ul style="list-style-type: none"> • MaxScoreAction = reject; • UseCustomReply = yes. <p>You can specify only text part of the reply: "550 5.7.0 <Text>".</p> <p>Text must be enclosed in quotation marks if it contains white spaces.</p> <p><u>Default value:</u> ReplyMaxScore = "Dr.Web MailD: Message is rejected due to score limit exceed."</p>
<pre>GetIpFromReceivedHeader = {logical}</pre>	<p>Instructs MailD core to use Received header value as a client's IP address if this address is not identified by Receiver.</p> <p>Note that in some cases Receiver cannot define client's IP address based on the analysis of Received header.</p> <p><u>Default value:</u> GetIpFromReceivedHeader = Yes</p>
<pre>Control = {logical}</pre>	<p>Enables drweb-maild (component MailD core) interactive management.</p> <p><u>Default value:</u> Control = No</p>
<pre>ControlAddress = {socket address}</pre>	<p>Socket address used by interactive management subsystem of drweb-maild module.</p> <p><u>Default value:</u> ControlAddress = inet:3009@127.0.0.1</p>
<pre>ControlPoolOption = {pool options}</pre>	<p>Thread pool settings for the interactive management socket of the drweb-maild.</p> <p><u>Default value:</u> ControlPoolOption = auto</p>
<pre>SkipDSNOnBlock = {logical}</pre>	<p>Skip DSN dispatch when the program failed to pass return code to Receiver after applying Reject or Tempfail actions.</p> <p><u>Default value:</u> SkipDSNOnBlock = No</p>

Features of Lookup usage in the ProtectedNetworks parameter

Lookups that retrieve Network IP address from a data source by the domain name or user name (that is, Lookups that use the \$d and \$u macros) cannot be set as a value of this parameter, because at this step, when the parameter is accessed to check the **SessionRestrictions** = trust_protected_network restriction, only the IP address, from which the connection was



established, is available and the address cannot be resolved in the FQDN.

For example, if you use such `Lookup` in the following SQL query

```
select net from networks where domain='$d'
```

the `net` address will not be selected from the database and, therefore, will not be marked as trusted.

However, you may use `Lookups` that retrieve Network IP addresses by the full address (that is, `Lookups` that use the `$s` macro). At that, the `Lookup` points to the client's IP address, so it can be used only in queries to those data sources that either contain lists with IP addresses or can resolve an IP address in the FQDN. The following example shows a correct query:

```
select net from networks where net='$s'
```

If the `net` field contains a client's IP address that was inserted into the query via the `$s` macro, the IP address is marked as trusted.

Note that the `SkipDomains` setting does not work in a `Lookup`, as at this step the domain name is unidentified.

For details on restrictions concerning the usage of domain and user name, see [Lookup description](#).

Features of error processing

If during message processing an error or event matching one of the constrains (`MaxScore`, `MaxMimeParts`, `LicenseLimit`) occurs, the action specified in the corresponding parameter is applied:

- `EmptyFrom`
- `MaxScoreAction`
- `LicenseLimit`
- `ProcessingErrors`

Please be careful when specifying [actions](#) for these parameters. Remember that:

1. If one of the `discard`, `reject` or `tempfail` actions is specified, mail processing stops and a message is deleted without being delivered to its recipient. When the `discard` action is selected, the sender does not receive notifications if the message was rejected. Alternatively to `discard` action, `reject` and `tempfail` actions enable notifications to the sender upon message rejection. Depending on the operation mode, the sender can be notified with an SMTP response sent by **Receiver** (in the [synchronous](#) mode) or with a DSN sent by **Sender** (in [asynchronous](#) mode).
2. If the `pass` action is specified, mail processing also stops, but the message is transmitted for delivery without completion of the processing (that is, if some plug-ins did not check the message, they will not be called). So, if the event occurred before the message was saved to the storage (while the message was processed by plug-ins assigned to the `BeforeQueue` queue), the message will be delivered in the synchronous mode, otherwise (when the message was processed by plug-ins assigned to the `AfterQueue` queue) – in the asynchronous mode.
3. For the `EmptyFrom` parameter, the action `pass` cannot be specified. Instead of this action, you can specify `continue`. That starts message processing (because this event can occur only before the message is processed).



Please note that if the mandatory action of the `ProcessingErrors` parameter is `discard`, `reject` or `tempfail`, do not set a small value to the `IpcTimeout` (the [\[General\] section](#)) parameter, because checking of message content can take considerable time. Occurrence of timeout before the check completes is regarded as an error. Thus, in accordance with the action specified in `ProcessingErrors`, the message is to be deleted during its processing, which can lead to loss of the message: the message will not be delivered to the recipient and the sender will not be informed on that.



[MailBase] Section

In the [MailBase] section, settings for **Dr.Web MailD** internal database are specified. The database stores received messages until they are [processed](#) and sent if the processing [plug-ins](#) operate in the asynchronous mode. The section contains the following parameters:

MaxStoredMessages = {numerical value}	<p>Maximum number of messages to be stored in the mail database.</p> <p>When the parameter value is set to 0, maximum number of messages is not limited. If amount of messages in database exceeds the number set to this parameter, old messages are deleted until the specified quantity is reached. Already sent messages are deleted immediately, others are sent at first and then deleted.</p> <p><u>Default value:</u> MaxStoredMessages = 100000</p>
MaxStorageSize = {numerical value}	<p>Maximum size of the mail database (in bytes).</p> <p>When parameter value is set to 0, maximum size is not limited. If database size exceeds the limits, old messages are deleted until the specified size is reached (see description of the MaxStoredMessages parameter).</p> <p><u>Default value:</u> MaxStorageSize = 0</p>
MaxPoolSize = {numerical value}	<p>Maximum mail database pool size (maximum number of memory pages, size of a page – 8 KB).</p> <p>If the parameter value is set to 0, the pool size is set up automatically according to available physical memory.</p> <p>In the current version, this parameter cannot be changed when reloading by SIGHUP signal.</p> <p><u>Default value:</u> MaxPoolSize = 0</p>
SendTimeout = {time}	<p>Timeout for plug-in to perform an asynchronous scan of a message.</p> <p>When timeout is exceeded, it is assumed that an error occurred during the message check. Actions to be applied in this case are specified in the ProcessingErrors parameter of the [MailD] section.</p> <p><u>Default value:</u> SendTimeout = 30s</p>
FrozenTimeout = {time}	<p>Additional time for message processing.</p> <p>If the time specified in the SendTimeout parameter is not enough for a plug-in to process a message, the time can be extended by the value of the FrozenTimeout parameter.</p> <p>Note that this parameter is deprecated and is not used anymore.</p> <p><u>Default value:</u> FrozenTimeout = 2h</p>
DeleteTimeout = {time}	<p>Maximum time period for a message to be stored in the mail database.</p>



	<p>Default value: DeleteTimeout = 48h</p>
BackupPeriod = {time}	<p>Time period for a mail database backup. When the parameter value is set to 0, no backup is performed.</p> <p>Default value: BackupPeriod = 0</p>
BackupName = {path to file}	<p>Name of mail database backup file. If the specified file name ends with question mark ("?"), each backup is stored in a separate file, and the question mark in the file name is replaced with the time value when the copy is created.</p> <p>Default value: BackupName = %var_dir/msgs/db/.maildb.backup</p>
MaxBodySizeInDB = {size}	<p>Maximum size of a message body stored in mail database. When size of a message body exceeds this parameter value, the message is stored in a separate external file.</p> <p>Default value: MaxBodySizeInDB = 1k</p>
SyncMode = {logical}	<p>Synchronization mode used for internal database.</p> <p>If the value of this parameter is set to Yes, the <code>fsync</code> function is called for each transaction. As a result, the database stored on disk is always up-to-date, but system performance is decreased (in some cases only slightly).</p> <p>If the value is set to No, OS buffering is used for database synchronization. As a result, on <code>drweb-maild</code> module emergency shutdown, data from the last transactions can be lost, but the database will not be destroyed and system performance will increase.</p> <p>If there are no special requirements for system reliability, you may leave No as this parameter value.</p> <p>Default value: SyncMode = No</p>

[Notifier] Section

In the [Notifier] section, settings of `drweb-notifier` module (**Notifier** component) are specified. **Notifier** is responsible for generation and sending MailD notifications, DSNs and reports on operation of **Dr.Web for UNIX mail servers** components.

PoolOptions = {pool options}	<p>Notifier thread pool settings</p> <p>Default value: PoolOptions = auto</p>
TemplatesBaseDir = {path to directory}	<p>Path to the directory where templates of MailD notification, reports and DSNs are stored.</p> <p>Default value: TemplatesBaseDir = %etc_dir/maild/templates</p>
LngBaseDir = {path to directory}	<p>Path to the directory where language files used for generation of notifications are stored.</p>



		Description of language file structure and use is provided here .
		Default value: LngBaseDir = %etc_dir/mailed/lng
AdminMail = {email address}	R C	Postmaster email address. It is possible to specify several addresses. In this case, generated notifications and reports (except for DSN which is always dispatched only to the sender of a message which cannot be delivered) are sent to all specified addresses. At that, all of the addresses appear in the message body. It is recommended to configure this parameter, otherwise MailD notifications and reports will not be sent. Please note that you can set several values to this parameter. Default value: AdminMail = root@localhost
FilterMail = {email address}	R	email address specified in the From header of MailD notifications and reports. Please note that DSN is always sent with an empty sender's address field (that is required by standards). Thus, this parameter is not used for DSN dispatching. Default value: FilterMail = root@localhost
NotifyLangs = {string}	R C	Language(s) used for generation of notifications and reports. Listed languages must correspond to language files located in the directory specified in LngBaseDir . Description of language file structure and their usage is provided here . Plug-ins always use the first language in this list Default value: NotifyLangs = en
TemplatesParserLogLevel = {log level}		Log verbosity level of the subsystem that create reports based on templates. The following levels are allowed: <ul style="list-style-type: none">• quiet• error• alert• info• debug Default value: TemplatesParserLogLevel = info
RulesLogLevel = {log level}		Log verbosity level of the Rule processor The followed levels are allowed: <ul style="list-style-type: none">• quiet• error• alert• info• debug



	<p>Default value: RulesLogLevel = info</p>
<p>MsgIdMap = {string}</p>	<p>Mapping of message identifier set in Receiver to the identifier set in Sender that will receive notifications generated for the message.</p> <p>Format of mapping: <regular_expression> <Sender_ID></p> <p>where <regular_expression> – regular expression to which Receiver ID corresponds, and <Sender_ID> – ID of Sender which is notified.</p> <p>If mapping is not found, all notifications are sent to the default Sender component (with an empty identifier).</p> <p>Example: MsgIdMap = id[12] sender_notifications</p> <p>In this case, reports on messages generated by Receiver components with id1 or id2 identifiers are sent to Sender with the sender_notifications identifier.</p> <p>This parameter is used when several pairs of Sender and Receiver components operate simultaneously.</p> <p>Default value: MsgIdMap =</p>
<p>QuarantinePrefix = {string}</p>	<p>Prefix added to the output path to a file in Quarantine</p> <p>This parameter allows accessing quarantined files using the third-party server.</p> <p>For example, if you install an HTTP-server on the same host where Dr.Web for UNIX mail servers runs and set the following value to the parameter</p> <p>QuarantinePrefix = http://mailhost/quarantine/</p> <p>this prefix will be included in paths to quarantined files, for example, http://mailhost/quarantine/headersfilter/ drweb.quarantine.2kqtvI.</p> <p>Default value: QuarantinePrefix =</p>

For details on notifications, notification templates, and notification generation rules, refer to the [Notification Templates](#) section.

[Quarantine] Section

In the [Quarantine] section, settings for proper **Quarantine** operation are specified:

<p>Path = {path to directory}</p>	<p>Path to the Quarantine directory.</p> <p>Default value: Path = %var_dir/infected/</p>
<p>FilesMode = {permissions}</p>	<p>Permissions for files that are moved to Quarantine.</p> <p>Default value: FilesMode = 0660</p>



<p>FileNamesMode = {Std Tai Rand48}</p>	<p>Naming convention for files to be moved to Quarantine:</p> <ul style="list-style-type: none">• Std – renaming quarantined files with <code>mkstemp</code> command. %FileNamesPrefix.XXXXXX template is used, where %FileNamesPrefix is the prefix specified in the FileNamesPrefix parameter value and XXXXXX is a combination of random letters and digits;• Tai – renaming quarantined files according to TAI (International Atomic Time). %sec.%usec.%FileNamesPrefix.XXXXXX template is used;• Rand48 – renaming quarantined files with <code>lrand48</code> command. %FileNamesPrefix.XXXXXXXXXX template is used. <p><u>Default value:</u> FileNamesMode = Std</p>
<p>FileNamesPrefix = {string}</p>	<p>Prefix used to rename files which are moved to Quarantine. The parameter value must not contain "%", "/" and "_" characters.</p> <p><u>Default value:</u> FileNamesPrefix = maild</p>
<p>AccessByEmail = {logical}</p>	<p>Permission to process requests to receive messages saved to Quarantine via control messages.</p> <p>Control message must be sent to the email address specified in the FilterMail parameter value (or in Rules) with the special Subject header:</p> <pre>q:relative_path_to_file</pre> <p>where <code>relative_path_to_file</code> is a relative path to the quarantined file (for example, <code>/drweb/drweb.quarantine.puYtWx</code>). Corresponding message is sent in response to such request only if one of its recipients or its sender matches control message sender.</p> <p>Such control message is automatically generated by MUA of the MailD notification recipient when the corresponding link in the received report is clicked.</p> <p>Please note that the default value of the OnlyTrustedControlMails parameter of the [Maild] section is <code>Yes</code>, thus control messages must be sent from a protected network (specified in the ProtectedNetworks parameter of the [Maild] section). Otherwise, the control message is ignored.</p> <p><u>Default value:</u> AccessByEmail = Yes</p>
<p>StoredTime = {time}</p>	<p>Period of time to store a message in Quarantine. When the parameter value is set to 0, this period of time is not limited.</p> <p><u>Default value:</u> StoredTime = 24h</p>
<p>MaxSize = {size in Kbytes}</p>	<p>Maximum total size of messages in Quarantine, in KB. If value of this parameter is set to 0, the size is not limited.</p>



	<p>For each message, size of the message body is calculated rather than its actual size on the disk.</p> <p>This parameter affects only the size of internal database and does not affect the DBI storage (if connected).</p> <p>Default value: MaxSize = 0</p>
MaxNumber = {numerical value}	<p>Maximum number of messages in Quarantine.</p> <p>If value of this parameter is set to 0, this number is not limited.</p> <p>This parameter affects only the number of messages in the internal database and does not affect the DBI storage (if it is connected).</p> <p>Default value: MaxNumber = 0</p>
MoveToDBI = {Yes No}	<p>Moving of quarantined messages from file storage to the DBI storage.</p> <p>To move messages to the DBI storage, the <code>File::Temp</code> and DBI Perl modules are required.</p> <p>Default value: MoveToDBI = No</p>
DBISettings = {string}	<p>DBI storage connection parameters.</p> <p>Example: "dbi:Pg:dbname=emails_db"</p> <p>Database must be created using SQL-ASCII character set.</p> <p>Requirements to format of the table used for message storing are presented below.</p> <p>Default value: DBISettings =</p>
DBIUsername = {text value}	<p>User name to connect to the DBI storage.</p> <p>Default value: DBIUsername =</p>
DBIPassword = {text value}	<p>User password to connect to the DBI storage.</p> <p>Default value: DBIPassword =</p>
SQLInsertCommand = {string}	<p>An SQL command to add a message to the DBI storage.</p> <p>Sequence of fields listed in the command must correspond to the format of the table in DBI (see below). Inserted values in the request must be replaced with question marks ("?").</p> <p>SQL command must contain the following fields:</p> <ul style="list-style-type: none">• Message number• Relative path to a file with message. File format is the following: <code>client/plugin/id.prefix</code>, where <code>client</code> 'def' string, <code>plugin</code> – name of the plug-in, <code>id</code> – message number in hexadecimal form (in output, the first eight symbols are used), <code>prefix</code> is the prefix depending on values of the following parameters: FilenameMode и FilenamePrefix• Time of saving a message to the database



	<ul style="list-style-type: none">• Value of the <code>From:</code> header (enclosed in angle brackets)• List of recipients' addresses. Addresses in the list are separated by commas and enclosed in angle brackets• Message body. <p>Example:</p> <pre>SQLInsertCommand = "INSERT INTO mail_export(id, filename, put_time, sender, rcpts,body) values (?,?,?, ?, ?, ?)"</pre> <p>Default value:</p> <pre>SQLInsertCommand =</pre>
<pre>SQLRemoveCommand = {string}</pre>	<p>A command to delete messages from the DBI storage.</p> <p>It is used when time limit for storing messages in Quarantine is specified. The only parameter specified in request is <code>time</code>, all messages older than the specified value are deleted.</p> <p>The value element in the request must be replaced with a question mark ("<code>?</code>").</p> <p>Example:</p> <pre>SQLRemoveCommand = "DELETE FROM mail_export WHERE put_time<=?"</pre> <p>Default value:</p> <pre>SQLRemoveCommand =</pre>
<pre>SQLSelectCommand = {string}</pre>	<p>A command used to access messages in the DBI storage (for example, to request a message from Quarantine using control message).</p> <p>The only parameter used in the request is a relative file name in Quarantine. Value element in request must be replaced with a question mark ("<code>?</code>").</p> <p>Sequence and types of returned fields are fixed (corresponds to the format of the table used in the storage, see below):</p> <ol style="list-style-type: none">1. <code>id</code> - message number2. <code>put_time</code> - time of saving message to the database3. <code>body</code> - message body4. <code>sender</code> - value of the <code>From:</code> header (enclosed in angle brackets)5. <code>rcpts</code> - list of recipients' addresses. Addresses in the list are separated by commas and enclosed in angle brackets6. <code>filename</code> - relative path to the file with message <p>Example:</p> <pre>SQLSelectCommand = "SELECT id,put_time,body,sender,rcpts,filename FROM mail_export WHERE filename LIKE ?"</pre> <p>Default value:</p> <pre>SQLSelectCommand =</pre>
<pre>PulseTime = {time}</pre>	<p>Period of time to delete old messages and move messages from the DBI storage.</p> <p>When the parameter value is set to 0, the program specified in the <code>PathToDrwebQp</code> parameter value is not started.</p> <p>Default value:</p> <pre>PulseTime = 5m</pre>



PathToDrwebQp = {path to file}	Path to the drweb-qp utility. Default value: PathToDrwebQp = %bin_dir/drweb-qp
MoveAll = {logical}	Move all incoming messages directly to the / Path_parameter_value/def/backup/ directory and then archive them. The parameter must be used with MoveToDBI = Yes, otherwise the directory can quickly become full with incoming messages. Default value: MoveAll = No

Format of database table used for storing quarantined messages

Table in **DBI** used for message storing must contain the following fields (order of the fields is not important, but their names and types must be identical to those specified below):

- **id** (number) – Message number (identifier).
- **filename** (string) – Relative path to the file with the message.
- **put_time** (timestamp) – Time of adding the message to the database.
- **sender** (string) – Value of the **From:** header (enclosed in angle brackets).
- **rcpts** (string) – List of recipients from the message header (**TO:**, **CC:**, **BCC:**). Values are separated by commas and enclosed in angle brackets.
- **body** (string) – Message body.

Please note that "data types" presented in the list, must be replaced with similar data types that are available in the used DBMS (*integer*, *varchar* and others).

[Filters] Section

In the [Filters] section, general setting of **Dr.Web MailD plug-ins** are specified:

LibDir = {path to directory}	A directory with plug-ins. Default value: LibDir = %bin_dir/mauld/plugins/
Settings = {list of plug-in settings}	Plug-ins startup options. They are specified in the following format: <plugin_settings>, <plugin_settings>... where <ul style="list-style-type: none"> • <plugin_settings> is a string plugin_name: <PARAM> ... <PARAM> • <PARAM> is a pair setting_name = setting_value. Example: Settings = vaderetro: max_size = 400k log_level=debug, drweb: max_size = 10m For Vaderetro plug-in : maximum size of a message to scan is set to 400 Kbytes, log verbosity level is set to debug; for Drweb plug-in : maximum size of a message to scan is set to 10 Mbytes. A full list of plug-in options that can be specified in the Settings parameter is presented below . All values (except for file paths and file names in UNIX) are case-



	<p>insensitive.</p> <p><u>Default value:</u> Settings =</p>
<p>BeforeQueueFilters = {list of plug-ins}</p>	<p>List of plug-ins which process a message before it is queued or moved to the database (in the synchronous mode 'before-queue').</p> <p><u>Default value:</u> BeforeQueueFilters =</p>
<p>MaxSizeBeforeQueueFilters = {size}</p>	<p>Maximum size of a message to be processed by plug-ins that are listed in the BeforeQueueFilters parameter.</p> <p>Used only when the max_size parameter value is not explicitly specified for a plug-in (in the Settings parameter or Rules).</p> <p>When the parameter value is set to 0, maximum size is not limited.</p> <p><u>Default value:</u> MaxSizeBeforeQueueFilters =</p>
<p>AfterQueueFilters = {list of plug-ins}</p>	<p>List of plug-ins which process a message after it is put to the queue or to the database (in the asynchronous mode 'after-queue').</p> <p><u>Default value:</u> AfterQueueFilters =</p>
<p>MaxSizeAfterQueueFilters = {size}</p>	<p>Maximum size of a message to be processed by plug-ins that are listed in the AfterQueueFilters parameter value.</p> <p>Used only when the max_size parameter value is not explicitly specified for a plug-in (in the Settings parameter or Rules).</p> <p>When the parameter value is set to 0, maximum size is not limited</p> <p><u>Default value:</u> MaxSizeAfterQueueFilters = 0</p>
<p>PluginsBaseDir = {path to directory}</p>	<p>Path to the directory where plug-ins' work files are stored.</p> <p>For example, Vaderetro plug-in searches for the file of the used VadeRetro library in this directory</p> <p><u>Default value:</u> PluginsBaseDir = %var_dir/plugins/</p>

In the current version, only the following parameters can be specified for plug-ins. Specify the plug-in name in the **Settings** parameter in the following way: <plugin_name>:<parameter> . If the parameter is used in [Rules](#) of message processing, specify as follows: <plugin_name>/<parameter>.

<p>section = {text value}</p>	<p>Name of the configuration file section where plug-in parameters are specified (as a rule, a plug-in configuration file consists of one section).</p> <p>If the section name is not specified, the section with the plug-in name is used.</p>
<p>max_size = {size}</p>	<p> Maximum size of a message to scan.</p> <p>When the parameter value is set to 0, maximum size is not limited.</p> <p>Its default value depends on the queue (BeforeQueueFilters</p>



	<p>or AfterQueueFilters) which the plug-in is assigned to, and is defined by the value of the MaxSizeBeforeQueueFilters or MaxSizeAfterQueueFilters parameters.</p> <p>Using the parameter in Rules (in the configuration file):</p> <pre>[Rules] #Rule that is true for all messages true cont plugin_name/max_size = {size}</pre> <p>Example:</p> <pre>[Rules] ... #For email messages from admin@domain.com set max_size of the Drweb plug-in to 100k from:admin@domain.com cont drweb/max_size = 100k</pre>
<pre>log_level = {log level}</pre>	<p>R Log verbosity level, used for plug-ins</p> <p>The following levels are allowed:</p> <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug <p>Default value of this parameter is the same as the Level parameter value in the [Logging] section.</p>
<pre>log_ipc_level = {log level}</pre>	<p>R IPC library log verbosity level.</p> <p>The following levels are allowed:</p> <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug <p>Default value of this parameter is the same as the IpcLevel parameter value from the [Logging] section.</p>
<pre>syslog_facility = {syslog label}</pre>	<p>R Log type label which is used by <code>syslogd</code> system service.</p> <p>Default value of this parameter is the same as the SyslogFacility parameter value in the [Logging] section.</p>
<pre>log_filename = {syslog path to file}</pre>	<p>Path to the log file name.</p> <p>You can specify <code>syslog</code> as a log file name and logging will be carried out by <code>syslogd</code> system service.</p>
<pre>path_to_lib = {path to file}</pre>	<p>R Path to the dynamic-link library of a plug-in, if the library name does not correspond to <code>lib<plugin_name>.so</code> naming rules or the library is not located in the directory specified in the LibDir parameter.</p> <p>Path to the library can be absolute or relative. The relative path is specified from the directory set in the LibDir parameter</p> <p>Default value:</p> <pre>path_to_lib = LibDir/lib<plugin_name>.so</pre>



If most part of message traffic consists of large messages (with large attachments or with a large number of attachments), these messages are processed by plug-ins for a long time. In this case, it is not recommended to assign plug-ins to the **BeforeQueueFilters** list as it slows down interaction with MTA when transmitting messages.

Moreover, in this case errors can occur if the **IpTimeout** parameter value (the [General] [section](#)) is too small (less than the average time period required for message processing). That might result in loss of messages (they will not be delivered to the recipient, and the sender will not be informed about that).

[Rule] Section

Parameter values that are frequently used in [Message processing rules](#) (SETTINGS [part](#)) can be organized into named groups. Each of the groups is present in the main **Dr.Web MailD** configuration file as a section:

```
[Rule: <group name>]
```

where <group name> – unique name of the setting group. The name is case insensitive and can contain Latin characters, numbers and white spaces.

Parameter values in each section are specified as <Parameter> = <Value> pairs, one parameter per line (that is why it is not required to escape commas in parameter values). Section ends when another section of the configuration file starts (any section including one with settings of another named group) or when the configuration file ends.

Settings specified in a named group can be used for any mail processing Rule with the **rule=<group name>** directive (see [Rules of message processing](#)). In the current version of **Dr.Web MailD**, you can specify no more than 1 rule parameter. Number of user sections is not restricted.



All sections that contain named setting groups must be specified before the [Rules] section in the configuration file.

Example:

These lines declare `MySettings` section which specifies two parameters (block MailD notifications and disable moving to **Quarantine**, see description below):

```
[Rule:MySettings]
quarantine = no
notify = block
```

The following two rules, specified in the configuration file, use the section to set values of the quarantine and notify parameters:

```
[Rules]
Rcpt:regex:example\.com cont      rule=MySettings
Sender:lol@foo.com && block:vir1 cont notify.Skip=allow, notify.Virus=allow,
rule=MySettings
```

After that, MailD notifications are blocked and messages sent to the **example.com** domain cannot be moved to **Quarantine**. If a message is sent from the `lol@foo.com` address and the `vir1` blocking object is found, MailD notifications on detected viruses and skipped messages will be allowed (for all types of notification recipients). At that, other notifications and moving files to **Quarantine** are blocked; these settings are imported from the used `MySettings` group (specified in the [Rule:MySettings] section).

Section of default settings

Main configuration file always includes section with default settings for those parameters that are not specified in the file - their values can be set [only in Rules](#). This setting group has strictly defined name



default. Thus, the section which contains settings in this group has the following heading: `[Rule: default]`, at that, `default` can be omitted and the section name can be as follows: `[Rule]`. To use default settings in a rule, specify `rule=default` directive.



Do not confuse the `[Rule]` section and the `[Rules]` section: in the latter, [Rules of message processing](#) are specified!

The `[Rule]` section contains default values for all parameters that can be used in Rules. See description of these parameters in the chapter [Parameters used in Rule SETTINGS](#).

Example:

```
[Rule]
notify                = block
notify.Virus          = allow(any)
notify.Cured          = allow(admin:sender)
notify.Skip           = block
notify.Archive        = allow(admin)
notify.Error          = allow(admin)
notify.Rule           = allow(admin)
notify.License        = allow(admin)
notify.Malware        = allow(any)
html = yes
```

In the given example, default values for the `notify` and `html` parameters are set. They will be used for all email messages unless the values are redefined by a Rule.

[Rules] Section

The `[Rules]` section of the main configuration file contains general rules of message processing.



Note that the `[Rules]` section structure differs from structure of other sections: instead of `<Parameter> = <Value>` pairs it contains Rules of message processing that are specified one per line (if force hyphenation is not used).

For details on the format used for specifying a rule, see [Rules of message processing](#).

Parameters that must be used for processing of a certain message are searched downwards according to the order they are specified in the `[Rules]` section. Thus, the order you set them is important: more specific rules must be mentioned before more general rules (for details, see [Message processing](#)).

If specified Rules are associated with users (that is, they have sender's or recipient's email addresses as a condition) and number of users with individual rules is high, using of the rules for setting specific processing parameters is ineffective, as complexity of search is proportional to the number of the Rules. Therefore, it is recommended to store individual user settings in the [local database](#). In this case, search of true rules is performed more effectively and moreover, memory usage is optimized.

Parameter values that are frequently used in Message processing rules (`SETTINGS` part) can be organized into [named groups](#). Each of the groups with `<NAME>` name is present in the main **Dr.Web MailD** configuration file as a section `[Rule: <NAME>]`.

Note that all sections with named setting groups must be specified above the `[Rules]` section.



[Stat] Section

In the [Stat] section, parameters of gathering statistics on **Dr.Web MailD** operation are specified:

<pre>Detail = {off low medium high}</pre>	<p>You can specify one of the following verbosity levels for statistics gathering:</p> <ul style="list-style-type: none">• <code>off</code> - disables statistics gathering, which allows to increase performance of the suite. As a result, statistics cannot be exported and no reports are sent.• <code>low</code> - enables gathering of statistics on operation of the whole suite. As a result, statistics can be exported and reports are sent.• <code>medium</code> - to the statistics gathered on the <code>low</code> level, statistics on groups is added. You can enable or disable statistics collection for a certain group.• <code>high</code> - to the statistics gathered on the <code>medium</code> level, statistics on each user listed in the internal database is added. You can enable or disable statistics gathering for a certain user. <p>You can access statistics via the control socket or the web interface. Statistics collected on the <code>low</code> level is also included in reports (if this option is enabled).</p> <p><u>Default value:</u> Detail = <code>low</code></p>
<pre>Send = {logical}</pre>	<p>Sending reports to the statistics server (or Dr.Web Control Center if Dr.Web MailD is working as a part of the Anti-virus network in central protection mode).</p> <p><u>Default value:</u> Send = <code>Yes</code></p>
<pre>SendPeriod = {time}</pre>	<p>Time interval to send statistics to server.</p> <p><u>Default value:</u> SendPeriod = <code>10m</code></p>
<pre>Timeout = {time}</pre>	<p>Timeout for the statistics server to response.</p> <p><u>Default value:</u> Timeout = <code>30s</code></p>

It is possible to export statistics with **Dr.Web MailD** using the [Storage](#) type.

To enable export of statistics via the [Storage type](#):

1. Specify `Yes` as a value of the **ExportStat** parameter.
2. Configure at least one of the following parameters accordingly:

<pre>ExportStat = {logical}</pre>	<p>Export statistics to storages specified in the corresponding parameters (see below).</p> <p><u>Default value:</u> ExportStat = <code>No</code></p>
<pre>ExportBlockObjectsStorage = {string}</pre>	<p>List of parameters that configure export of statistics on blocked messages.</p> <p>This data is saved immediately after a message is blocked but only if it was scanned by the anti-virus module (that is, statistics is not exported if a message was blocked because of processing errors).</p>



Names of the table and fields in the database can be arbitrary but they must be of the same type as exported data. Fields in the query must be ordered like in the database.

It is not necessary to use all available values in the query.

Text fields (<varchar_long>) must be enclosed in single quotation marks(').

List of values that can be used in the query:

- :number<int> - unique message identifier;
- :q_name<varchar_long> - path to the quarantined file where the message was saved (if it was saved in **Quarantine**);
- :virus_name<varchar_long> - name of the blocked object found in the message;
- :virus_code<int> - code of the blocked object found in the message.

The list of codes:

- 1 - infected;
 - 2 - virus modification;
 - 3 - suspicious;
 - 4 - cured;
 - 5 - deleted;
 - 6 - filtered;
 - 7 - skipped;
 - 8 - archive restrictions;
 - 9 - errors;
 - 10 - read errors;
 - 11 - write errors;
 - 12 - adware;
 - 13 - dialer;
 - 14 - joke;
 - 15 - riskware;
 - 16 - hacktool.
- :plugin_name<varchar_long> - name of the plugin which blocked the message;
 - :sender<varchar_long> - sender's address enclosed in angle brackets;
 - :client_ip<varchar_long> - IP address of the client that loaded the message into the mail database (if available);
 - :date<timestamp> - timestamp of loading the message into the mail database;
 - :client_id<varchar_long> - the unique identifier of the **Client** for which saving into the mail database is performed (always 'def' string).

Example:

```
ExportBlockObjectsStorage = "odbc:insert into
viruses values (:number<int>,
':q_name<varchar_long>',
':virus_name<varchar_long>', :virus_code<int>,
':plugin_name<varchar_long>',
':sender<varchar_long>',
':client_ip<varchar_long>', :date<timestamp>,
':client_id<varchar_long>')"
```



	<p><u>Default Value:</u> ExportBlockObjectsStorage =</p>
<p>ExportStatStorage = {string}</p>	<p>Export of statistics on number of processed messages. Export is performed:</p> <ul style="list-style-type: none">• on shutdown;• after a period of time specified in the SendPeriod parameter value. <p>If statistics is empty (no messages were processed), nothing is exported.</p> <p>Names of the table and fields in the database can be arbitrary but they must be of the same type as exported data. Fields in the query must be ordered like in the database.</p> <p>It is not necessary to use all available values in the query.</p> <p>List of values that can be used in the query:</p> <ul style="list-style-type: none">• :size<int> - total size of scanned messages in bytes;• :num<int> - total number of scanned messages;• :q_num<int> - total number of messages saved in Quarantine;• :r_num<int> - total number of redirected messages;• :n_num<int> - total number of messages for which notifications were sent;• :pass_num<int> - total number of passed messages;• :reject_num<int> - total number of rejected messages;• :discard_num<int> - total number of discarded messages;• :tempfail_num<int> - total number of temporarily failed messages;• :date<timestamp> - mail database timestamp;• :q_size<int> - total size of messages saved in Quarantine;• :r_size<int> - total size of redirected messages;• :n_size<int> - total size of messages for which notifications were sent;• :pass_size<int> - total size of passed messages;• :reject_size<int> - total size of rejected messages;• :discard_size<int> - total size of discarded messages;• :tempfail_size<int> - total size of temporarily failed messages;• :work_time<int> - plug-in operation time, in milliseconds (ms). <p>Example:</p> <pre>ExportStatStorage = "odbc:insert into q_stat values(:size<int>, :num<int>, :q_num<int>, :r_num<int>, :n_num<int>, :pass_num<int>, :reject_num<int>, :discard_num<int>, :tempfail_num<int>, :date<timestamp>)"</pre>
	<p><u>Default value:</u> ExportStatStorage =</p>



ExportPluginStatStorage = {string}

Export of statistics on number of processed messages by each of the plug-ins. Statistics are exported only for plug-ins that are specified in the **Names** parameter value of the [Reports] [section](#) (or for all working plug-ins if the parameter value is not specified). Export is performed:

- on shutdown;
- on SIGHUP signal;
- when sending a report to Administrator;
- after a specified period of time, if reports are not sent too often.

If statistics are empty (no messages were processed), nothing is exported.

Names of the table and fields in the database can be arbitrary but they must be of the same type as exported data. Fields in the query must be ordered like in the database.

List of values that can be used in the query:

- the same as for the **ExportStatStorage** parameter;
- :plugin_name<varchar_long> - name of the plug-in for which statistics are exported;

Example:

```
ExportPluginStatStorage = "odbc:insert into
plugin_stat values(':plugin_name<varchar_long>',
:size<int>, :num<int>, :q_num<int>, :r_num<int>,
:n_num<int>, :pass_num<int>, :reject_num<int>, :
discard_num<int>, :tempfail_num<int>, :date<time
stamp>)"
```

Default value:

ExportPluginStatStorage =

For details on the statistics export options, see [Exporting Statistics](#).

[Reports] Section

In the [Reports] section, parameters that configure creation and sending of [plug-in](#) activity reports are specified:

Send =
{logical}

Enables or disables sending of reports.

Default value:

Send = Yes

SendTimes =
{schedule}

Schedule for sending reports.

The syntax is as follows:

- hour:minute:second[-period] – send a report at the specified time every day;
- Nw/hour:minute:second[-period] – send a report at the specified time on the Nth day of week (0 – Sunday, 1 – Monday, 2 – Tuesday and so on);
- Nm/hour:minute:second[-period] – send a report at the specified time on Nth day of month.

If the period of time is specified (period), report includes data for this period, otherwise a report includes data for 24 hours.

Example:

```
SendTimes = 00:00:00-24h, 1w/00:00:00-7d,
2M/21:23:32-31d
```



	<p>In this case, three reports are sent: daily report at midnight, weekly report on Monday midnight and monthly report on the second day of each month at 21:23:32.</p> <p><u>Default value:</u> SendTimes = 24h</p>
<p>Mail = {email address}</p>	<p>email addresses where reports are sent.</p> <p>If the parameter value is not specified, email addresses defined in the AdminMail parameter value in the [Notifier] section are used. You can specify several email addresses separated by commas.</p> <p>Please note that if the value of the Mail parameter is set, reports are not sent to the address specified in the AdminMail parameter.</p> <p><u>Default value:</u> Mail =</p>
<p>Names = {list of plug-ins}</p>	<p>List of plug-ins, for which report is created. The plug-ins are separated by commas.</p> <p>If this parameter value is not specified, report is created for plug-ins listed in the BeforeQueueFilter and AfterQueueFilter parameters in the [Filters] section.</p> <p><u>Default value:</u> Names =</p>
<p>TopListSize = {numerical value}</p>	<p>Outputs to the report the lists of frequently blocked objects and addresses, from which maximum amount of blocked objects was sent.</p> <p>The parameter value defines number of entries in each list. If the parameter value is set to 0, lists are not created. If the parameter value is set to -1, size of the lists is not limited.</p> <p><u>Default value:</u> TopListSize = 20</p>
<p>MaxStoreInDbPeriod = {time}</p>	<p>Maximum period of time to store statistics in the reports database.</p> <p>If the parameter value is set to 0, old entries are not deleted.</p> <p>Note that this parameter is deprecated and is not used anymore.</p> <p><u>Default value:</u> MaxStoreInDbPeriod = 31d</p>
<p>CheckForRemovePeriod = {time}</p>	<p>Period of time at the end of which old entries are deleted from the reports database.</p> <p>Note that this parameter is deprecated and is not used anymore.</p> <p><u>Default value:</u> CheckForRemovePeriod = 5m</p>

[Logging] Section

In the [Logging] section, parameters for [log files](#) are specified. Logging is performed for all main **Dr.Web for UNIX mail servers** modules.

<p>Level = {log level}</p>	<p>Log verbosity level used by Dr.Web MailD components.</p>
---------------------------------------	--



	<p>The following levels are available:</p> <ul style="list-style-type: none"> • Quiet • Error • Alert • Info • Debug <p><u>Default value:</u> Level = Info</p>
<pre>IPCLevel = {log level}</pre>	<p>Log verbosity level of the IPC library.</p> <p>The following levels are available:</p> <ul style="list-style-type: none"> • Quiet • Error • Alert • Info • Debug <p><u>Default value:</u> IPCLevel = Alert</p>
<pre>SyslogFacility = {syslog label}</pre>	<p>Log type label which is used by the <code>syslogd</code> system service.</p> <p><u>Default value:</u> SyslogFacility = Mail</p>
<pre>FileName = {syslog path to file}</pre>	<p>Name of the log file or <code>syslog</code>, if the syslog system service is used</p> <p><u>Default value:</u> FileName = <code>syslog</code></p>

Using SASL

This chapter provides you with description of sections that contain SASL authentication parameters.

Current version of **Dr.Web for UNIX mail servers** supports SASL authentication only via the `cyrus-sasl` driver. Thus, if SASL authentication is used, both `[SASL]` and `[Cyrus-SASL]` sections must exist in the configuration file.

See also a [configuration example](#) of authentication via **Cyrus SASL**.

[SASL] Section

In the `[SASL]` section, you can specify parameters of SASL authentication in **Dr.Web for UNIX mail servers** (designed for working as a proxy-server via SMTP/LMTP protocol):

<pre>Use = {logical}</pre>	<p>Enables or disables SASL authentication.</p> <p><u>Default value:</u> Use = No</p>
<pre>Driver = {text}</pre>	<p>Name of the used SASL authentication driver.</p> <p>In the current version, only the <code>cyrus</code> driver is available. To use it, install and set up <code>cyrus-sasl2</code> library.</p>



	<p><u>Default value:</u> Driver = cyrus</p>
BrokenAuthClients = {logical}	<p>Support for outdated SMTP clients which use irregular syntax of AUTH protocol.</p> <p><u>Default value:</u> BrokenAuthClients = Yes</p>
AuthenticatedHeader = {logical}	<p>Adds names of registered users to the Received header.</p> <p>When the value is set to Yes, names of registered users are visible to everyone.</p> <p><u>Default value:</u> AuthenticatedHeader = No</p>

[Cyrus-SASL] Section

In the [Cyrus-SASL] section, parameters which configure operation of `cyrus-sasl` SASL driver are specified:

Lib = {path to file}	<p>Absolute path to the <code>cyrus-sasl2</code> library.</p> <p><u>Default value:</u> Lib = /usr/lib/libsasl2.so.2</p>
Path = {string}	<p>Name of the configuration file (.conf extension is added automatically).</p> <p>The <code>cyrus-sasl2</code> library receives its settings from this file.</p> <p>Note that Dr.Web MailD does not check whether the file, specified in the parameter, exists and is correct. If the file is missing or is incorrect, the <code>cyrus-sasl2</code> library automatically uses its default settings and no notification is generated on that.</p> <p><u>Default value:</u> Path = maild</p>
ServerHostname = {string}	<p>Host name.</p> <p>FQDN that is automatically added as @domain to the <code>user</code> part transmitted by a client (if only the <code>user</code> part - UID - is transmitted). The result string is used as a login that <code>saslauthd</code> searches for authorization.</p> <p>If the value is not set, the Hostname parameter value from the [General] section is used instead. If Hostname value is also not specified, value returned by <code>gethostname</code> function is used.</p> <p><u>Default value:</u> ServerHostname =</p>
ServerRealm = {string}	<p>SASL realm the server belongs to.</p> <p>FQDN that is automatically added as @domain to the <code>user</code> part of address transmitted by a client. The result string <code>user@domain</code> is used as a login that <code>saslauthd</code> searches for authorization.</p> <p>If the value is not set and the client did not transmit <code>Realm/Domain</code> value when authenticating, FQDN is taken from the ServerHostname parameter.</p> <p>Note that you need to specify <code>-r</code> option for the <code>saslauthd</code></p>



	daemon to enable correct processing of the <code>user@domain</code> string while authentication.
	Default value: ServerRealm =
SecurityOptions = {string}	List of security settings, separated by commas. The following security settings are allowed: <ul style="list-style-type: none"> • <code>noplaintext</code> – disable authentication mechanisms susceptible to attacks (for example, PLAIN, LOGIN); • <code>noactive</code> – protection from active (non-dictionary) attacks during authentication exchange; • <code>nodictionary</code> – disable authentication mechanisms susceptible to passive dictionary attacks; • <code>noanonymous</code> – disable authentication mechanisms that allow anonymous login; • <code>mutual_auth</code> – require mutual authentication.
	Default value: SecurityOptions = <code>noanonymous</code>

MTA Connections

This section provides you with description of parameters that configure interaction between **Dr.Web for UNIX mail servers** and different MTAs.

The configuration file must contain sections with parameters which specify interaction between **Dr.Web for UNIX mail servers** and used MTA. If **Dr.Web for UNIX mail servers** operates in [mail filtering mode](#), the configuration file must also contain the `[POP3]` or `[IMAP]` section, depending on the used protocol.

Depending on the MTA, **Dr.Web MailD** uses [different modules](#) that perform functions of **Sender** and **Receiver** components. The following table lists MTAs that can be integrated with **Dr.Web for UNIX mail servers**. For every MTA, the corresponding modules and sections of the configuration file are specified.

MTA	Operation mode	Modules used as Receiver and Sender and corresponding sections	
Sendmail (or any MTA that uses Milter protocol)	—	<code>drweb-milter</code>	[Milter]
		<code>drweb-sender</code>	[Sender]
Postfix	before-queue	<code>drweb-receiver</code>	[Receiver]
		<code>drweb-sender</code>	[Sender]
	after-queue	<code>drweb-receiver</code>	[Receiver]
		<code>drweb-sender</code>	[Sender]
	Milter	<code>drweb-milter</code>	[Milter]
		<code>drweb-sender</code>	[Sender]
Exim	—	<code>drweb-receiver</code>	[Receiver]
		<code>drweb-sender</code>	[Sender]
CommuntGate Pro	—	<code>drweb-cgp-receiver</code>	[CgpReceiver]
		<code>drweb-cgp-sender</code>	[CgpSender]
Courier	—	<code>drweb-courier</code>	[Courier]
		<code>drweb-sender</code>	[Sender]



MTA	Operation mode	Modules used as Receiver and Sender and corresponding sections	
Zmailer	—	drweb-zmailer drweb-sender	[Receiver] [Sender]
Qmail	—	drweb-qmail drweb-sender	[Qmail] [Sender]
SMTP/LMTP proxy, default	—	drweb-receiver drweb-sender	[Receiver] [Sender]

For details on integration between **Dr.Web for UNIX mail servers** with MTA, see [Integration with Mail Transfer Systems](#).

[Receiver] Section

In the [Receiver] section, settings of the **Receiver** component (module **drweb-receiver**) are specified. This component is used in **Dr.Web for UNIX mail servers**, if the solution interacts with **Exim**, **Zmailer** and **Postfix** mail systems (if **Postfix** mail system does not use the **Milter** protocol) or operates in the **SMTP/LMTP proxy** mode.

1. General parameters

Address = {address}	Address used by Receiver to receive messages. Socket used for receiving messages is specified (either TCP socket, or UNIX socket). Default value: Address = inet:25@0.0.0.0
PoolOptions = {pool options}	Thread pool settings for Receiver . Default value: PoolOptions = auto
RealClients = {logical}	Accept connections directly from clients. Default value: RealClients = Yes
ProcessingErrors = {action}	Action applied to messages when processing errors occur. The parameter value can be one of the following actions : tempfail, discard, reject. Only one action can be specified. Default value: ProcessingErrors = reject
StalledProcessingInterval = {time}	Timeout to process stalled messages. Stalled messages are messages got by Receiver , but not processed in time and thus not sent to MailD core . That can happen when problems with network or power supply occur. If a stalled message is found, Receiver queues it for processing. Default value: StalledProcessingInterval = 10m
OneCommandTimeout = {time}	Timeout to execute a single command.



	<p>Default value: OneCommandTimeout = 5m</p>
OneMessageTimeout = {time}	<p>Timeout to receive a single message.</p> <p>Default value: OneMessageTimeout = 10m</p>
AddReceivedHeader = {logical}	<p>Adds a Received header to all received messages.</p> <p>Default value: AddReceivedHeader = Yes</p>
ReturnReject = {logical}	<p>Receiver behaviour after the Reject action is applied to the message that is processed in the synchronous mode.</p> <p>When the parameter value is set to Yes, the component returns SMTP 55* error. When the parameter value is set to No, the component returns successful SMTP 250 response, but the sender receives DSN (if not disabled).</p> <p>The return response is extended with the Reply<Reason> string (the string is set in the settings of a plug-in which performed the reject action), but only if it is allowed by its UseCustomReply=Yes setting (where <Reason> is the reject action cause). Otherwise, the following standard message is output: "The message has been rejected by the Dr.Web MailD".</p> <p>Note that if Dr.Web MailD is not operating in SMTP/LMTP proxy mode and is integrated with MTA, it is recommended to set the parameter value to No. This ensures correct notification of the sender if a message was rejected. Otherwise, if the parameter value is set to Yes, MTA can send notification on success before the message is checked and rejected.</p> <p>If ReturnReject=No, it is recommended to specify an additional notify action (as during SMTP session Receiver, after rejecting a message, replies with 250 code that indicates successful message processing), or enable DSN (with the SkipDSNOnBlock parameter in the [Maild] section). But it is recommended to enable DSN only if the number of rejected messages is not large; otherwise, high load on MTA, which sends DSN, can occur.</p> <p>Default value: ReturnReject = Yes</p>
GreetingString = {string}	<p>Greeting line that is output on connection of a new SMTP client.</p> <p>"%host%" macro is replaced with the Hostname parameter value from the [General] section.</p> <p>"%ver%" macro is replaced with the current version of the drweb-receiver module.</p> <p>Default value: GreetingString = "%host% Dr.Web SMTP receiver v% ver% ready"</p>
RelayDomains = {Lookup}	<p>List of domains that are allowed to relay messages.</p> <p>If you specify a usual domain list, for which Dr.Web MailD is a mail relay, their subdomains are ignored. That is, mail arriving from their subdomains are not relayed.</p>



	<p>It is possible to specify a list of subdomains by using regular expression or <code>rfile</code>.</p> <p>Please note that the parameter value is Lookup.</p> <p>Example: <code>RelayDomains = regex:*.domain.com</code></p> <p>Allows relaying to all <code>domain.com</code> subdomains.</p> <p>Example: <code>RelayDomains = rfile:/path</code></p> <p><code>rfile</code> contains a list of regular expressions (Perl syntax), which must be specified one per line:</p> <pre>*.domain.com *.domain1.com *.domain2.com</pre> <p>In the current version of Dr.Web MailD, the <code>RelayDomains</code> parameter does not support wildcard DNS records. Thus, expressions of this type are not allowed:</p> <pre>RelayDomains = *.domain</pre> <p><u>Default value:</u> <code>RelayDomains =</code></p>
<pre>RestrictionStat = {logical}</pre>	<p>Enables or disables statistics on SMTP restrictions (description of restrictions is provided below).</p> <p>To get statistics, send <code>SIGUSER1</code> signal to the <code>drweb-receiver</code> process. Statistics is stored in the <code>restrictions.txt</code> file in the directory defined in the <code>BaseDir</code> parameter from the [General] section.</p> <p><u>Default value:</u> <code>RestrictionStat = No</code></p>
<pre>DelayRejectToRcpt = {logical}</pre>	<p>Suspends block of messages until <code>RCPT</code> stage, even if a restriction was applied before.</p> <p>Setting this parameter allows working with outdated versions of email clients and output the list of blocked recipient addresses to the log file.</p> <p><u>Default value:</u> <code>DelayRejectToRcpt = Yes</code></p>

2. Numerical restrictions of SMTP session

The following parameters allow setting numerical restrictions upon violation of which SMTP protocol dialog is aborted.

<pre>MaxRecipients = {numerical value}</pre>	<p>Maximum number of recipients for one email message (number of <code>RCPT TO</code> commands).</p> <p>When the parameter value is set to 0, maximum number of recipients is not limited.</p> <p>If an IP address from which connection is established is marked as <code>trusted</code>, this restriction is not checked.</p> <p><u>Default value:</u> <code>MaxRecipients = 100</code></p>
--	---



<code>MaxConcurrentConnection = {numerical value}</code>	<p>Maximum number of concurrent SMTP connections from a single IP address.</p> <p>When the parameter value is set to 0, maximum number of SMTP connections from a single IP address is not limited.</p> <p><u>Default value:</u> MaxConcurrentConnection = 5</p>
<code>MaxMailsPerSession = {numerical value}</code>	<p>Maximum number of messages per single session (number of MAIL FROM commands).</p> <p>When the parameter value is set to 0, maximum number of messages per single session is not limited.</p> <p><u>Default value:</u> MaxMailsPerSession = 20</p>
<code>MaxReceivedHeaders = {numerical value}</code>	<p>Maximum number of Received headers.</p> <p>When the parameter value is set to 0, maximum number of Received headers is not limited.</p> <p>Receiver always checks this restriction, even if an IP address is marked as <i>trusted</i>.</p> <p><u>Default value:</u> MaxReceivedHeaders = 100</p>
<code>MaxErrorsPerSession = {numerical value}</code>	<p>Maximum number of errors per single session.</p> <p>When the parameter value is set to 0, maximum number of errors per single session is not limited.</p> <p><u>Default value:</u> MaxErrorsPerSession = 10</p>
<code>MaxMsgSize = {size}</code>	<p>Maximum message size (transmitted in DATA command).</p> <p>Receiver always checks this restriction, even if an IP address is marked as <i>trusted</i>.</p> <p><u>Default value:</u> MaxMsgSize = 10m</p>
<code>MaxJunkCommands = {numerical value}</code>	<p>Maximum number of RSET, NOOP and VRFY commands per session.</p> <p>If this number exceeds the specified value, an error counter activates.</p> <p>Current value of the error counter is set to 0 each time the message is successfully processed by the drweb-maild module.</p> <p>If the parameter value is set to 0, this restriction is not checked.</p> <p><u>Default value:</u> MaxJunkCommands = 100</p>
<code>MaxHELOCommands = {numerical value}</code>	<p>Maximum number of HELO, EHLO and LHLO commands per session.</p> <p>If this number exceeds the specified value, an error counter activates. The score is reset after every successful processing of the message by drweb-maild.</p> <p>If the parameter value is set to 0, this restriction is ignored.</p>



Default value:
MaxHELOCommands = 20

Please note that some of the restrictions mentioned above are checked even if an IP addresses are marked as `Trusted`. The following table describes behavior of restrictions for `Trusted` clients and texts of SMTP replies that are sent if a message does not satisfy the restriction.

Restriction	Message to the sender if the connection to Client is restricted	Whether checked for trusted connections
MaxRecipients	452 4.5.3 Too many rcpts	No
MaxConcurrentConnection	421 4.7.0 Too many concurrent SMTP connections from this IP address; please try again later	No
MaxMailsPerSession	421 4.2.1 too many messages in this connection	No
MaxReceivedHeaders	554 5.7.0 MailD error: Too many received headers: N	Yes
MaxErrorsPerSession	421 4.7.0 Error: too many errors	No
MaxMsgSize	552 5.3.4 Message size exceeds file system imposed limit	Yes
MaxJunkCommands	421 4.7.0 Error: too many errors	No
MaxHELOCommands	421 4.7.0 Error: too many errors	No

3. Restrictions and conditions for different stages of SMTP session

Parameters described below (***Restrictions**) configure check of IP addresses on various SMTP session stages. Check is performed if the address is not marked as `Trusted`. By default, only connections from `localhost` and UNIX sockets are considered `trusted`.

The restrictions allow filtering of unwanted mail in `drweb-receiver` [module](#) on the stage of SMTP session, before messages are transmitted to `drweb-maild`. That saves resources and adds an additional level of spam filtration, which increases spam detection probability.

SMTP Restrictions are applied on the following stages of SMTP session:

- connection of the new client (INTRO) (restrictions are specified in the `SessionRestrictions` parameter);
- receipt of `HELO/EHLO` command (restrictions are specified in the `HeloRestrictions` parameter);
- receipt of `FROM` command – that is, when the client specifies sender for the new message (restrictions are specified in the `SenderRestrictions` parameter);
- receipt of `RCPT` command – that is, when the client adds a new recipient to the message (restrictions are specified in the `RecipientRestrictions` parameter);
- receipt of `DATA` command – that is, when the client has already finished transferring all recipients and is ready to send the body of the message (restrictions are specified in the `DataRestrictions` parameter).

Restrictions are set as values of ***Restrictions** parameters separated by commas. They are checked in sequential order – from left to right. Restriction checking is performed only after all other checks (sequencing of commands, validity of their parameters and others) until the message is considered as `trusted`. After that, restriction check stops.

SessionRestrictions = {restrictions list}

These checks are performed immediately after the connection was established (INTRO).



	<p>The following restrictions are checked:</p> <ul style="list-style-type: none">• trust_protected_network• trust_protected_domains• trust_white_networks• trust_white_domains• reject_dnsbl• reject_black_networks• reject_black_domains <p><u>Default value:</u> SessionRestrictions = trust_protected_network</p>
<p>HeloRestrictions = {restriction list}</p>	<p>These checks are performed on HELO/EHLO session stage.</p> <p>The following restrictions are checked:</p> <ul style="list-style-type: none">• reject_unknown_hostname• reject_diff_ip <p><u>Default value:</u> HeloRestrictions =</p>
<p>SenderRestrictions = {restriction list}</p>	<p>Checks performed on FROM session stage.</p> <p>The following restrictions can be checked:</p> <ul style="list-style-type: none">• reject_unknown_sndrs• reject_unknown_domain• trust_sasl_authenticated• pass_sasl_authenticated <p><u>Default value:</u> SenderRestrictions = trust_sasl_authenticated</p>
<p>RecipientRestrictions = {restriction list}</p>	<p>Checks performed on RCPT session stage. All recipients are checked in order they are declared.</p> <p>The following restrictions can be checked:</p> <ul style="list-style-type: none">• reject_unknown_domain• reject_unauth_destination• reject_unknown_rcpts• pass_sasl_authenticated <p><u>Default value:</u> RecipientRestrictions = reject_unauth_destination</p>
<p>DataRestrictions = {restriction list}</p>	<p>Checks performed on DATA stage of session.</p> <p>The following restrictions can be checked:</p> <ul style="list-style-type: none">• reject_spam_trap• reject_multi_recipient_bounce• pass_sasl_authenticated <p><u>Default value:</u> DataRestrictions =</p>

Result of blocking can be different depending on the stage of SMTP session. When blocking is performed according to restrictions from the **SessionRestrictions** parameter (INTRO stage) – the whole session is blocked: that is, an error is returned on any user command. On other SMTP stages, only a certain SMTP command is blocked.



Each restriction can have an optional parameter – score value [SCORE] (except for `set_score` and `add_score` restrictions, where the score value is the only mandatory parameter). Depending on the restriction type, score is processed in different ways:

- restriction can be applied if the current message score is less than the value specified in the parameter
- restriction can be applied if the current message score is greater than the value specified in the parameter
- if restriction is applied, the corresponding parameter value is added to the message score.

Depending on the SMTP session stage, restrictions can affect either a score that is added to the one of every message in the current session (for `SessionRestrictions` and `HeloRestrictions` stages) or an individual message score (for other stages).

Each stage of the check could have its own specific restrictions as well as restrictions that are actual for all stages. The latter include the following restrictions:

Action	Description
<code>sleep {time}</code> [SCORE]	Suspend the SMTP connection for the specified period (in <code>seconds</code>) If SCORE is specified, this restriction is applied only to messages the current score of which is greater than the parameter value.
<code>reject</code> [SCORE]	Return the permanent SMTP error (code 5*). If SCORE is specified, the permanent error is returned only when the current message score is greater than the parameter value.
<code>tempfail</code> [SCORE]	Return the temporary SMTP error (code 4*). If SCORE is specified, then temporary error is returned only when the current message score is greater than the parameter value.
<code>mark_trust</code> [SCORE]	Set <code>Trusted</code> flag. All other restrictions after this parameter are to be skipped. If SCORE is specified, <code>Trusted</code> flag is set only when the current message score is lower than the parameter value.
<code>set_score</code> SCORE	Changes the current message score to the specified SCORE value. If it is used on <code>SessionRestrictions</code> or <code>HeloRestrictions</code> stages, it affects the score of every message in the session, on other stages it affects the score of the current processed message.
<code>add_score</code> SCORE	Add the specified SCORE value to the current message score. If it is used on <code>SessionRestrictions</code> or <code>HeloRestrictions</code> stages, it affects the score of every processed message in the session, on other stages it affects the score of the current processed message.

Restrictions actual for different check steps:

Action	Description
Actions for <code>SessionRestrictions</code>	
<code>trust_protected_network</code> [SCORE]	If the IP address of the connection is included in the list specified in the <code>ProtectedNetworks</code> parameter (the [Maild] section), the address is either marked as <code>Trusted</code> or, if SCORE is specified, its value is added to the score of each message transferred in the current session and to the score of the sender's IP address.
<code>trust_protected_domains</code> [SCORE]	Checks if the IP address of the connection is in the list defined by the <code>ProtectedDomains</code> parameter (the



Action	Description
	<p>[Maild] section).</p> <p>The check is performed using double DNS request: PTR request is sent to check if the received host name is in the ProtectedDomains list. If so, an A request is sent to check if the connection IP address is in the received address list. If so, address is either marked as <code>Trusted</code> or, if <code>SCORE</code> is specified, its value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
<code>trust_white_networks</code> [SCORE]	<p>If the IP address of the connection is in the white list defined by the WhiteNetworks parameter (see below), the address is either marked as <code>Trusted</code> or, if <code>SCORE</code> is specified, its value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
<code>trust_white_domains</code> [SCORE]	<p>Checks if the domain of the IP address is in the white list defined by the WhiteDomains parameter (see below).</p> <p>DNS PTR request is made. If the domain is in the list, the address is either marked as <code>Trusted</code> or, if <code>SCORE</code> is specified, its value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
<code>reject_dnsbl</code> [SCORE]	<p>Checks if the IP address of the connection is in the black lists of RBL/DNSBL servers specified in the DNSBLList parameter (see below).</p> <p>At first, availability of RBL/DNSBL servers is checked by sending a test request to resolve <code>127.0.0.2</code> IP address (as required by the specification). If a server operates correctly, it must return positive response. If not, the server is marked as inaccessible, which is logged.</p> <p>If the server is available, an A request to DNSBL is sent.</p> <p>If the DNSBL server sent a positive response, the session terminates or, if <code>SCORE</code> is specified, its value is added to the score of each message transferred in the current session and to the score of the sender's IP address and the error is logged.</p> <p>Please note that if all specified RBL/DNSBL servers are inaccessible, the IP address is considered "untrusted", reject_dnsbl is not applied to it and a record that all of the servers are unavailable is logged.</p>
<code>reject_black_networks</code> [SCORE]	<p>If the IP address of the connection is in the black list defined by the BlackNetworks parameter (see below), the session terminates or, if <code>SCORE</code> is specified, an error is logged, and the <code>SCORE</code> value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
<code>reject_black_domains</code> [SCORE]	<p>Checks if the sender's domain is in the black list defined by the BlackDomains parameter (see below).</p> <p>PTR request is made. If the domain is in this list, the session terminates or, if <code>SCORE</code> is specified, an error is logged, and the <code>SCORE</code> value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
Actions for <code>HelRestrictions</code> restriction	



Action	Description
<code>reject_unknown_hostname</code> [SCORE]	<p>If the host name has neither DNS A record nor DNS MX record, mail from this address is blocked or, if SCORE is specified, an error is logged, and the SCORE value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p> <p>During check, A requests and, sometimes, MX requests are sent.</p>
<code>reject_diff_ip</code> [SCORE]	<p>If the client's IP address does not match any of the IP addresses resolved for the domain name from the EHLO/HELO command, mail from this address is blocked.</p> <p>If SCORE is specified, the message is passed, but an error is logged and the SCORE value is added to the score of each message transferred in the current session and to the score of the sender's IP address.</p>
Actions for SenderRestrictions	
<code>reject_unknown_domain</code> [SCORE]	<p>If the sender's host name has neither DNS A record nor DNS MX record, mail from this address is blocked or, if SCORE is specified, an error is logged, and the SCORE value is added to the score of each message transferred in the current session.</p> <p>During check, A requests and sometimes MX requests are sent.</p> <p>It is recommended to use <code>reject_unknown_domain</code> together with other restrictions for this session stage (<code>reject_unknown_domain</code> restriction is checked first and then - the others). It is necessary because if the FROM field of email message is empty, this restriction is not applied (as there is no domain name to check in DNS). But it is impossible to ban email messages with empty FROM and TO fields because, according to RFC 5321 specification, Dr.Web MailD must always be able to receive DSN and MSN notifications that have empty FROM fields <>.</p>
<code>trust_sasl_authenticated</code> [SCORE]	<p>If SASL authentication was successful, the IP address is marked as <code>Trusted</code>. However, if SCORE is specified, it is also checked whether the IP address score is less than the specified value.</p>
<code>pass_sasl_authenticated</code> [SCORE]	<p>Skip all other checks on this SMTP session stage if the client has successfully passed SASL authentication.</p> <p>If SCORE is specified, a client that has successfully passed SASL authentication is checked unless the current score is less than the specified parameter value.</p>
<code>reject_unknown_sndrs</code> [SCORE]	<p>Checks if the recipient is specified in the <code>ProtectedSenderEmails</code> list (see below).</p> <p>If the sender's address is not in this list, mail from this address is blocked or, if SCORE is specified, an error is logged and the SCORE value is added to the message score.</p> <p>It is recommended to use this action together with <code>anti_dha</code> Reputation IP Filter.</p>
Actions for RecipientRestrictions	
<code>reject_unknown_domain</code> [SCORE]	<p>If the recipient host name has neither DNS A record nor DNS MX record, mail to this address is blocked or, if SCORE is specified, an error is logged, and the SCORE value is added to the message score.</p>



Action	Description
	<p>During check, A requests and, sometimes, MX requests are sent.</p> <p>Recommendations on how to use <code>reject_unknown_domain</code> together with other restrictions for this session stage are similar to the recommendations described above for <code>SenderRestrictions</code> stage, but they are actual for <code>TO</code> field.</p>
<code>reject_unauth_destination</code> [SCORE]	<p>If the recipient's domain is neither in the <code>RelayDomains</code> list nor in the <code>ProtectedDomains</code> list (see the [Maild] section), mail sent to this address is blocked or, if <code>SCORE</code> is specified, an error is logged, and the <code>SCORE</code> value is added to the message score.</p> <p>If mail for subdomains of protected domains is also received (that is, the <code>IncludeSubdomains</code> parameter is set to <code>Yes</code>), it is required to set both <code>reject_unauth_destination</code> and <code>reject_unknown_domain</code> restrictions on the RCPT stage. Otherwise, Dr.Web MailD will receive messages for all subdomains of the protected domains even if these subdomains do not exist.</p>
<code>reject_unknown_rcpts</code> [SCORE]	<p>Checks if the recipient is in the <code>ProtectedEmails</code> list (see below).</p> <p>If the recipient's address is not in this list, mail to this address is blocked. If <code>SCORE</code> is specified, an error is logged, and the <code>SCORE</code> value is added to the message score.</p> <p>It is recommended to use this action with anti_dha Reputation IP Filter.</p>
<code>pass_sasl_authenticated</code> [SCORE]	<p>Skip all other checks on this SMTP session stage if the client has successfully passed SASL authentication.</p> <p>If <code>SCORE</code> is specified, a client that has successfully passed SASL authentication is checked unless the current score is less than the specified parameter value.</p>
Actions for DataRestrictions restriction	
<code>reject_spam_trap</code> [SCORE]	<p>Checks for a spam trap. The recipient's address must be of the <code><USER@HOST></code> format.</p> <p>If the host name is in the list defined by the <code>ProtectedDomains</code> parameter (unless the list is empty, see below) and the user name is in the list defined by the <code>SpamTrap</code> parameter (see below), the message is blocked or, if <code>SCORE</code> is specified, an error is logged and the <code>SCORE</code> value is added to the message score. Full email address can be also specified in the <code>SpamTrap</code> list.</p>
<code>reject_multi_recipient_bounce</code> [SCORE]	<p>Blocks messages with empty FROM header and several recipients or, if <code>SCORE</code> is specified, an error is logged and the <code>SCORE</code> value is added to the message score.</p>
<code>pass_sasl_authenticated</code> [SCORE]	<p>Skip all other checks on this SMTP session stage if the client has successfully passed SASL authentication.</p> <p>If <code>SCORE</code> is specified, a client that has successfully passed SASL authentication is checked unless the current score is less than the specified parameter value.</p>

**Examples:**

```
SenderRestrictions = trust_protected_networks, reject
```

Allows receiving email messages only from IP addresses which belong to the networks specified in the **ProtectedNetworks** parameter value. Messages from other IP addresses are rejected.

```
SenderRestrictions = trust_protected_networks, trust_protected_domains, sleep 5,
add_score 10
```

Allows receiving email messages from IP addresses which belong to the networks specified in the **ProtectedNetworks** parameter value and from domains specified in the **ProtectedDomains** parameter value. Processing of other messages is paused for 5 seconds and 10 points are added to the score of these messages.

It is possible to gather [statistics on each restriction](#) to define the quantity of blocked messages and efficiency of the restriction. To get the gathered data, send the special signal to the **drweb-receiver** process as described in the [Signals](#) section of the current manual. To enable or disable statistics gathering, use the **RestrictionStat** parameter (see above, in part 1 of this section).

4. Check parameters for different SMTP sessions

```
BlackNetworks =
{Lookup}

WhiteNetworks =
{Lookup}
```

Network black and white lists.

These lists are used in `trust_white_networks` and `reject_black_networks` actions.

Syntax is similar to the one of the **ProtectedNetworks** parameter in the [\[Maild\] section](#).

Please note that the parameter value is [Lookup](#).

Default value:

```
BlackNetworks =
WhiteNetworks =
```

```
DNSBLList =
{LookupLite}
```

DNSBL server list. This list is used in `reject_dnsbl` action.

Servers are checked one after another in the order they are specified in the parameter value until the message is blocked (upon the server response that the IP address is a "spammer") or the list ends.

Accessibility of the servers is checked by requesting them to resolve special IP address 127.0.0.2 (defined by the specification). The server must return positive response. Otherwise, the server is marked as unavailable, which is logged.

If no server responds to the request, the IP address is considered absent in the list of DNSBL servers, that is, the IP address is considered "clean".

Please note that the parameter value is [LookupLite](#).

Default value:

```
DNSBLList =
```

```
PositiveDNSBLCacheTimeout = {time}
```

Maximum time for caching positive responses from DNSBL servers.

Default value:

```
PositiveDNSBLCacheTimeout = 24h
```

```
NegativeDNSBLCacheTimeout = {time}
```

Maximum time for caching negative responses from DNSBL servers.



	<p>Default value: NegativeDNSBLCacheTimeout = 10m</p>
NegativeDNSCacheTimeout = {time}	<p>Maximum wait time for caching negative responses from DNS servers.</p> <p>Parameter value is valid for all responses from DNS servers except for those from DNSBL servers.</p> <p>Default value: NegativeDNSCacheTimeout = 10m</p>
BlackDomains = {Lookup} WhiteDomains = {Lookup}	<p>Black and white lists of domains. These lists are used in <code>trust_white_domains</code> and <code>reject_black_domains</code> actions.</p> <p>Syntax is similar to the one of the ProtectedDomains parameter in the [Maild] section.</p> <p>Please note that the parameter value is Lookup.</p> <p>Default value: BlackDomains = WhiteDomains =</p>
SpamTrap = {LookupLite}	<p>Spam trap address list.</p> <p>This list is used in <code>reject_spam_trap</code> action.</p> <p>Please note that the parameter value is LookupLite.</p> <p>Default value: SpamTrap =</p>
ProtectedEmails = {Lookup}	<p>List of protected recipients' addresses.</p> <p>It is used in <code>reject_unknown_rcpts</code> restriction. It allows to discard messages with invalid recipients (that are not in the list) and to resist DHA attacks (when used with <code>anti_dha</code> filter in Reputation IP Filter).</p> <p>It is recommended to specify this parameter with <code>reject_unknown_rcpts</code> restriction and use it with <code>anti_dha</code> filter.</p> <p>Please note that the parameter value is Lookup.</p> <p>Default value: ProtectedEmails =</p>
ProtectedSenderEmails = {Lookup}	<p>List of protected senders' addresses.</p> <p>It is used in <code>reject_unknown_sndrs</code> restriction. It allows to discard messages from invalid (unknown) senders and to resist DHA attacks (when used with <code>anti_dha</code> filter in Reputation IP Filter).</p> <p>It is recommended to specify this parameter together with <code>reject_unknown_sndrs</code> and use it with <code>anti_dha</code> filter.</p> <p>Please note that the parameter value is Lookup.</p> <p>Default value: ProtectedSenderEmails =</p>
ReputationIPFilter = {filters list}	<p>Reputation IP Filter settings.</p> <p>Reputation IP filter allows assigning a score to the IP address according to the gathered statistics on connections as well as</p>



	<p>blocking this IP address temporarily if its total score is greater than some threshold value.</p> <p>The following filters are available:</p> <p>anti_dha, errors_filter, score_filter.</p> <p>Filters are listed using comma as a delimiter, and are checked in order they were specified. For each filter, its name is specified first and then optional parameters are enumerated with a comma as a delimiter.</p> <p><u>Default value:</u></p> <p>ReputationIPFilter =</p>
MaxSessionScore = {numerical value}	<p>A threshold value for the general score of each session.</p> <p>If this score exceeds the threshold value, the corresponding connection will be closed and a temporary error is returned.</p> <p>If this value is set to 0, this parameter is ignored.</p> <p><u>Default value:</u></p> <p>MaxSessionScore = 10000</p>

[Sender] Section

In the [Sender] section, settings of the **Sender** component (responsible for sending messages) are specified. This section is not included in **Dr.Web** distribution for operation with **CommuniGate Pro** mail transfer system.

UseSecureHash = {logical}	<p>Instructs to add the X-DrWeb-Hash header ("mark") to messages that are sent back to the mail system.</p> <p>The parameter value is used if Dr.Web MailD has to reject an original message after modifying it and then add the modified message to the queue of incoming email. In this case, a message with such header, after being received by Dr.Web MailD from the mail system, is transmitted for delivery without a repeated check; otherwise, a message without the header is checked. Repeated checks of messages can cause looping and result in failure to deliver.</p> <p>The parameter must be used while interaction with the following mail systems: Postfix, Qmail, Sendmail, and Zmailer:</p> <ul style="list-style-type: none">• When Dr.Web MailD is interacting with the Postfix mail system, Yes must be specified only if the Milter protocol is used (interaction via the drweb-milter module). If so, drweb-milter processes all messages generated by drweb-sender. <u>Default value:</u> No• When Dr.Web MailD is interacting with the Qmail or Sendmail system, Yes must be specified if messages are sent and received by the same mail system. <u>Default value:</u> Yes• When Dr.Web MailD is interacting with the Zmailer system, Yes must be specified only if drweb-zmailer is used on the routing stage (e.g., started from process.cf). In this case, drweb-zmailer processes all messages generated by drweb-sender. <u>Default value:</u> No
-------------------------------------	--



<p>SecureHash = {string}</p>	<p>The parameter value determines content of the X-DrWeb-Hash header.</p> <p>You can specify a free-form string as a value, recommended string length is not less than 10 characters. For better security, it is strongly recommended to change the default parameter value.</p> <p>In Dr.Web for UNIX mail servers solution for Zmailer MTA this parameter value must be the same as the value of the <code>--hash parameter</code> used on <code>drweb-zmailer</code> startup, when Zmailer is used at the routing stage.</p> <p>Default value: SecureHash = !!!----- <u> </u> <code>EDIT_THIS</code> <u> </u> !!!</p>
<p>StalledProcessingInterval = {time}</p>	<p>Timeout for processing of stalled messages.</p> <p>Stalled message is a message that was received and processed but not transmitted to Sender for dispatch. This situation might occur due to network or power supply problems.</p> <p>If a stalled message is found, it is queued for dispatch.</p> <p>Also this parameter value is used for repeated requests to datasources from Lookups specified in the Router parameter (see below), if the used data sources are not accessible and therefore, a message cannot be delivered (as the destination address is unknown).</p> <p>Default value: StalledProcessingInterval = 10m</p>
<p>SendingIntervals = {time}</p>	<p>Time period between attempts to send stalled messages and notifications that were not sent on the first attempt.</p> <p>When Dr.Web for UNIX mail servers is running in synchronous mode, Sender attempts to send processed messages regardless of the first interval specified in the parameter value. If the attempt is unsuccessful, Sender starts dispatching delayed messages after a time period specified in the SendingIntervals parameter value. If the first parameter value is 0, it is ignored as an attempt to send a message was already performed.</p> <p>If Dr.Web for UNIX mail servers is running in asynchronous mode, Sender always attempts to send messages according to time intervals specified in the parameter value.</p> <p>Please note that all generated notifications and DSN are always sent in asynchronous mode (regardless of Dr.Web MailD operation mode). Thus, dispatch of delayed messages and DSN starts according to the first time interval specified in the SendingIntervals list. So, it is recommended to set 0 as the first interval.</p> <p>If the first value in the SendingIntervals list is not 0, in the asynchronous mode processed messages and notifications are sent with the delay equal to the first value in the list.</p> <p>If the SendingIntervals list have only one value, equal to 0, Sender does not attempt to send stalled messages. These messages are immediately moved to <code>/out/failed</code> directory. See also the warning at the end of this section.</p> <p>Default value: SendingIntervals = 0s, 30s, 60s, 10m, 30m, 2h, 8h, 1d, 1d</p>



Method = {SMTP LMTP PIPE}	Method used by Sender to deliver messages. <ul style="list-style-type: none">SMTP – messages are sent via SMTP protocol;LMTP – messages are sent via LMTP protocol;PIPE – messages are sent via PIPE to some external mail program. <p><u>Default value:</u> Depends on the distribution.</p>
MailerName = {SMTP Sendmail Postfix CommuniGate Qmail Exim Zmailer Courier}	Name of the MTA working with Dr.Web for UNIX mail servers . This parameter is used when Method = pipe. <p><u>Default value:</u> Depends on the distribution.</p>
Address = {address}	MTA address used by the Sender component to send messages. If Method = pipe, specify the full path to the MTA used to receive messages. If the Method parameter has another value, specify the address of the socket used for sending messages. If Dr.Web for UNIX mail servers solution operates in SMTP/LMTP proxy mode, then in addition to standard address types, you can use mx:HOSTNAME, where HOSTNAME is the name of the host. When this type is used, the suite receives all MX records and sends a message according to them. If only mx: prefix is specified, without a hostname, the software receives and uses MX records of the recipient's domain (from the TO filed in the message envelope). This parameter can have multiple addresses, separated by commas, for sending messages. Value of this parameter cannot be empty and must be specified even if routing is used (see description of the Router parameter). Example: Address = inet:25@10.4.0.90, inet:25@10.4.0.91, inet:25@10.4.0.92 In this example, if MTA with 10.4.0.90 address stops responding, Sender attempts to send an email message to 10.4.0.91. In case of unsuccessful transmission, the message is sent to 10.4.0.91. When number of addresses is large, it is recommended to increase values of the MaxTimeoutForThreadActivity and IpTimeout parameters (the [General] section) to at least 5 minutes , as to allow Sender to switch to the last address in case previous address the does not respond. <p><u>Default value:</u> Depends on the distribution.</p>
PipeTimeout = {time}	Timeout to receive response using pipe. <p><u>Default value:</u> PipeTimeout = 2m</p>
Options = {string}	Optional parameters for the external mail program, which is initialized when pipe method is used. <p><u>Default value:</u> Options =</p>



<code>InPoolOptions = {pool options}</code>	Thread pool settings for processing messages in 'before-queue' mode. Note that this parameter is deprecated and is not used anymore. Default value: <code>InPoolOptions = auto</code>
<code>OutPoolOptions = {pool options}</code>	Threads pool settings for processing messages in 'after-queue' mode. Default value: <code>OutPoolOptions = auto</code>

The following parameters are specified only in solutions for **Exim** and **Postfix** MTAs, and in **SMTP/LMTP proxy** mode.

<code>HeloCmdTimeout = {time}</code>	Timeout to execute HELO/EHLO commands. Default value: <code>HeloCmdTimeout = 5m</code>
<code>MailFromCmdTimeout = {time}</code>	Timeout to execute MAIL command. Default value: <code>MailFromCmdTimeout = 5m</code>
<code>RcptToCmdTimeout = {time}</code>	Timeout to execute RCPT command. Default value: <code>RcptToCmdTimeout = 5m</code>
<code>DataCmdTimeout = {time}</code>	Timeout to execute DATA/BDAT commands. Default value: <code>DataCmdTimeout = 2m</code>
<code>DataBlockTimeout = {time}</code>	Timeout to send a message. Default value: <code>DataBlockTimeout = 3m</code>
<code>EndOfDataTimeout = {time}</code>	Timeout to receive confirmation of message delivery. Default value: <code>EndOfDataTimeout = 10m</code>
<code>OtherCmdsTimeout = {time}</code>	Timeout to execute other commands via SMTP/LMTP. Default value: <code>OtherCmdsTimeout = 2m</code>
<code>SendDSN = {logical}</code>	Enables/disables sending of DSN reports if a message delivery problem occurs. Please note that if Dr.Web MailD operates via Sender with MTA (Exim , Postfix , Zmailer) and sending of DSN is enabled, it is necessary to be aware of the following situation. In this mode, DSN can be generated by Dr.Web MailD (after all attempts to send a message failed), and by the integrated MTA (if timeout for message processing expired). Thus, the sender can receive two



	<p>DNS about the problem (from Dr.Web MailD and from MTA).</p> <p>DSN is transmitted to Sender for delivery if it cannot be transmitted directly to MTA (for example, when the domain name in the sender address is not full) and the return code cannot be transmitted to Receiver.</p>
	<p><u>Default value:</u></p> <p>SendDSN = No</p>
<p>Router = {Lookup}</p>	<p>Message routing rules depending on recipients when the suite operates in SMTP/LMTP proxy mode.</p> <p>Messages addressed to different recipients can be sent via different routes. In this parameter, you can specify addresses for sending messages to different domains.</p> <p>Parameter values are specified in DOMAIN ADDRESS format, where:</p> <ul style="list-style-type: none">• DOMAIN is a string that must be included in receiver envelopes. Envelope has <user@host> format. Case-insensitive partial match is searched. For example, if "@localhost" substring is searched, <test@localhost> and <yy@localhost.localdomain> envelopes matches it, and if "@localhost>" substring is searched, only <test@localhost> envelope matches;• ADDRESS is an address where a message is sent to, if DOMAIN substring is found in the envelope. ADDRESS format is similar to the one of the Address parameter in this configuration file. It is possible to specify several email addresses delimited by the " " symbol, in this case, the message is delivered to the first address connection to which was established. <p>Note that the order in which addresses are specified is of importance because the first match is searched.</p> <p>If the matching DOMAIN string cannot be found for a message in the list, the address specified in the Address parameter is used for the message dispatch. Thus, the value of this parameter must not be empty (see a not below, in the Router usage subsection).</p> <p>Note that if data sources used in Lookup are inaccessible and the message cannot be delivered (as the destination address is unknown), this message is "lost" and Sender repeats an attempt to send the message at time interval specified in StalledProcessingInterval</p> <p>Example:</p> <pre>Router = @main.server.com> mx:main.server.com inet:25@backup.server.com</pre> <p>In this case messages addressed to recipients from main.server.com domain are sent to addresses indicated in MX record for main.server.com. If delivery fails, system tries to deliver the message to backup.server.com on port 25.</p> <p>Please note that the parameter value is Lookup.</p>
	<p><u>Default value:</u></p> <p>Router =</p>



Router usage

Router parameter allows using [Lookups](#) (except for `regex`, `wildcards` and `rfile` types).

Example:

```
Router = "mysql:select address from senders where user='$u'"
```

This query searches for a local part of the recipients address. The search is performed in `user` column of `senders` table in **MySQL** database. If the local part is found, the message is sent to the address specified in the matching row of the `address` column.

Example:

```
Router = "ldap:///description?sub?(cn='$d')", domain1.com inet:25@example.com |  
inet:1025@example.com | inet:2025@example.com, mail.com mx: | inet:25@mail.backup,  
domain2.com mx:mail.ru | inet:25@mail.backup, "file:/path/to/routers.list"
```

In this example, messages are transmitted for delivery the following way:

- 1) Recipient's domain name is searched using [LDAP](#) (`cn` attribute is used). If the domain name is found, redirection settings are taken from the `description` field.
- 2) Messages addressed to recipients from `domain1.com` are sent to `example.com` on port 25. If delivery fails, system tries to deliver the message to the same address on port 1025 and then on port 2025.
- 3) Messages addressed to `mail.com` are sent to addresses corresponding to MX record of `mail.com`.
- 4) Messages addressed to `domain2.com` are redirected to addresses specified for MX record of `mail.com` or to the port 25 of `mail.backup` server.
- 5) If address of the recipient does not match any of the previously described addresses, matching is checked in the `/path/to/routers.list` file.

Please note that only one address is assigned to one domain, therefore expressions of this type are not allowed:

```
Router = domain, domain2 25@host
```

Before using `Lookup` as a value of the **Router** parameter, check correctness of the `Lookup` (use [drweb-lookup utility](#)) and accessibility of the used data source. If the **Router** parameter value is `Lookup` with `OnError=exception` error handling mode (this mode can be set in data source settings as well as locally overridden in `Lookup` value expression), then if **Sender** cannot receive required route from the used data source, this situation must be handled as an error in **Sender**. An error message is logged. In this case:

- in the [synchronous](#) mode, **Receiver** returns 451 SMTP error code to the message sender (Requested action aborted: local error in processing), and the processed email message is removed from all queues of **Dr.Web MailD**.
- in the [asynchronous](#) mode, the message is marked as 'stalled' and **Sender** tries to send it at intervals specified in the `StalledProcessingInterval` parameter.

If an email message cannot be sent to any of the matching addresses, the following actions are performed (depending on the return code sent by the last MTA):

- **Sender** starts sending delayed messages after an interval specified in the `SendingIntervals` parameter. This process is configured by the rules specified in the **Router** parameter. If sending of delayed messages fails, **Notifier** generates a DSN (if allowed in the `SendDSN` parameter). If special routes are not specified for the domain of the undelivered message sender (the routes are specified in the **Router** parameter or in [message processing rules](#)), this DSN is sent to the address specified



in the **Address** parameter.

- if last MTA returns 5** SMTP code, DSN is generated immediately and the message is deleted from out-queues. Scheme of DSN sending is similar to the one described above. If DSN cannot be delivered, it is deleted after the interval specified in the **SendingIntervals** parameter.

When different routes are specified in the **Router** parameter and in [message processing rules](#) for the same domain, routes specified in the Rules have higher priority.



Please note that even if you configure routing of all mail messages with the **Router** parameter, the **Address** parameter value must not be empty; otherwise **Sender** fails to start. Address specified in the **Address** parameter value is used if the match in the routing table or in message processing rules cannot be found for the recipient.

If all attempts of **Sender** to dispatch a message are failed, the stalled message remains in the /out/failed directory until it is sent with **drweb-inject** [utility](#) or deleted manually (with the use of a system utility).

[Courier] Section

In the [Courier] section, settings of [interaction](#) between **Dr.Web MailD** and **Courier** mail transfer system are specified. This section is included in **Dr.Web MailD** configuration file only if the software version is designed for operation with the MTA mentioned above.

ProcessingTimeout = {time}	Timeout for the drweb-courier module to wait for a message to be scanned. It is recommended to set this parameter value greater than the value of the SendTimeout parameter from the [MailBase] section . Default value: ProcessingTimeout = 40s
ProcessingErrors = {action}	Action applied to messages that caused scan errors. Only one of these actions can be specified: tempfail, discard, pass, reject. Default value: ProcessingErrors = reject
MainPoolOptions = {pool options}	Settings for a pool of threads that process requests. Default value: MainPoolOptions = auto
ReplyPoolOptions = {pool options}	Settings for a pool of threads that process responses from the drweb-maild module. Default value: ReplyPoolOptions = auto
BaseDir = {path to directory}	Courier MTA installation directory. Default value: BaseDir = /usr/lib/courier
SocketDirs = {path to directory}	List of paths used to create UNIX sockets for interaction with the Courier MTA. UNIX socket is created in the first directory of the list, other



	<p>directories are checked for UNIX sockets with the same names as the <code>drweb-courier</code> module. Such UNIX sockets are deleted when found.</p> <p>In the current version this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required.</p> <p>Default value: SocketDirs = <code>/var/lib/courier/allfilters, /var/lib/courier/filters</code></p>
SocketAccess = {permissions}	<p>Permissions for UNIX socket files used for interaction between Dr.Web MailD and Courier MTA.</p> <p>In the current version this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required.</p> <p>Default value: SocketAccess = <code>0660</code></p>

[CgpReceiver] Section

In the `[CgpReceiver]` section, settings for [interaction](#) between **Receiver** and **CommuniGate Pro** mail transfer system are specified. This section is included in **Dr.Web MailD** configuration file only if the software version is designed for operation with the MTA mentioned above.

ProcessingTimeout = {time}	<p>Timeout for the Receiver component to wait for a message to be scanned.</p> <p>It is recommended to set this parameter value greater than the value of the <code>SendTimeout</code> parameter from the [MailBase] section.</p> <p>Default value: ProcessingTimeout = <code>40s</code></p>
PoolOptions = {pool options}	<p>Thread pool settings.</p> <p>Default value: PoolOptions = <code>auto</code></p>
ProcessingErrors = {action}	<p>Action applied to messages that caused scanning errors.</p> <p>Only one of these actions can be specified: <code>tempfail, discard, pass, reject</code>.</p> <p>Default value: ProcessingErrors = <code>reject</code></p>
ChownToUser = {string}	<p>Set owner for a message file received from CommuniGate Pro MTA.</p> <p>As <code>drweb-cgp-receiver</code> module runs with Administrator privileges (<code>root</code>), you can either leave this parameter value empty and start the whole Dr.Web for UNIX mail servers system with Administrator privileges, or you can set this parameter value to the name of a specific user whose privileges are used to run Dr.Web for UNIX mail servers (<code>drweb</code> by default).</p> <p>Default value: ChownToUser = <code>drweb</code></p>



[CgpSender] Section

In the [CgpSender] section, settings for [interaction](#) between **Sender** and **CommuniGate Pro** mail transfer system are specified. This section is included in **Dr.Web MailD** configuration file only if the software version is designed for operation with the MTA mentioned above.

<pre>UseSecureHash = {logical}</pre>	<p>Instructs to add the X-DrWeb-Hash header ("mark") to messages that are sent back to the mail system</p> <p>The parameter value is used if Dr.Web MailD has to reject an original message after modifying it and then add the modified message to the queue of incoming email.</p> <p>If No is specified, a message received by Dr.Web MailD from a mail system is transmitted directly for delivery, without performing a check, if the message was added to the mail system queue of incoming mail locally (via PIPE).</p> <p>If Yes is specified, a message received by Dr.Web MailD from a mail system is transmitted directly for delivery, without performing a check, if both the message has such header and the message was added to the mail system queue of incoming mail locally (via PIPE).</p> <p>Note that this parameter is used by both Sender and Receiver. Therefore, changing of the value requires not only sending of SIGHUP signal to Dr.Web Monitor (the signal makes Sender reread its configuration), but also restart of CGP mail system, which runs Receiver and makes the component reread the changed parameter value.</p> <p><u>Default value:</u> UseSecureHash = No</p>
<pre>SecureHash = {string}</pre>	<p>Content of X-DrWeb-Hash header.</p> <p>An arbitrary string of symbols (not less then 10 symbols) can be used as the parameter value. For better security, it is strongly recommended to change the default parameter value.</p> <p><u>Default value:</u> SecureHash = !!!----- __EDIT_THIS__ !!!</p>
<pre>PoolOptions = {pool options}</pre>	<p>Thread pool settings.</p> <p><u>Default value:</u> PoolOptions = auto</p>
<pre>SubmitDir = {path to directory}</pre>	<p>Directory where drweb-cgp-sender module submits messages for CommuniGate Pro MTA to send them.</p> <p><u>Default value:</u> SubmitDir = /var/CommuniGate/Submitted</p>
<pre>SubmitFilesMode = {permissions}</pre>	<p>Permissions for created notifications or saved messages.</p> <p><u>Default value:</u> SubmitFilesMode = 0600</p>
<pre>SubmitFileNamesPrefix = {string}</pre>	<p>Prefix for file names of submitted messages. File name format: %{SubmitDir}/%{SubmitFileNamesPrefix}XXXXXX</p> <p>It is possible to use "%s" macro which is replaced with a message identifier given to the message by CommuniGate Pro MTA and based on the file name. Usage of this macro can simplify log file</p>



	analysis.
	Default value: SubmitFileNamesPrefix = drweb_submit_%s_
SubmitFileNamesMode = {std tai rand48}	Naming convention for files of submitted messages: <ul style="list-style-type: none"> • Std – renaming files with <code>mkstemp</code> command. <code>drweb_submit_XXXXXX</code> template is used; • Tai – renaming files according to TAI (International Atomic Time). <code>%sec.%usec.drweb_submit_XXXXXX</code> template is used; • Rand48 – renaming files with <code>lrand48</code> command. <code>drweb_submit_XXXXXXXX</code> template is used.
	Default value: SubmitFileNamesMode = std

[Milter] Section

In the [Milter] section, parameters for managing operation of `drweb-milter` module are specified. This module is responsible for interaction between **Dr.Web for UNIX mail servers** and **Postfix** and **Sendmail** MTAs via `Milter` protocol. This section is included in the **Dr.Web MailD** configuration file only if the software version is designed for operation with MTAs mentioned above.

Address = {address}	Socket address to establish connection via <code>Milter</code> protocol. It must comply with definition specified in settings of mail system (in <code>sendmail.cf</code> configuration file of Sendmail MTA and in <code>main.cf</code> configuration file of Postfix MTA). Path to the PID file cannot be used as a value of this parameter. Example: Address = local:%var_dir/ipc/drweb-milter.skt In the current version this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required. Default value: Address = inet:3001@127.0.0.1
Timeout = {time}	Timeout for <code>drweb-milter</code> to connect to MTA. Specified value must be greater than any <code>Timeout</code> parameter value in the MTA configuration file. Default value: Timeout = 2h
PendedConnections = {numerical value}	Maximum queue length for pending connections (<code>drweb-milter</code> waits for MTA to process messages). Default value: PendedConnections = 64
CanChangeBody = {logical}	Enables MTA to modify the body of a message received from the mail system. Postfix MTA supports this function in version 2.4 or later. In the current version, this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required. If this parameter value is set to <code>Yes</code> , a checked message is



	<p>returned to MTA delivery queue with <code>drweb-milter</code> (Receiver) regardless of what queues (after-queue or before-queue) the checking plug-ins are assigned to.</p> <p>If this parameter value is set to <code>No</code>, a checked message is returned to MTA delivery queue with <code>drweb-sender</code> (Sender) if the message was modified during the check (for example, when a virus was removed), as in this case the message cannot be returned to the mail system queue and it is transmitted to MTA as a new message. If the message was not modified, it is returned to the MTA delivery queue with <code>drweb-milter</code> (Receiver) regardless of what queues the checking plug-ins are assigned to.</p> <p>All service notifications (including DSN), reports, redirected (with <code>redirect</code> action) and cloned messages are sent only with <code>drweb-sender</code> (Sender), regardless of the <code>CanChangeBody</code> parameter value and what queues the checking plug-ins are assigned to.</p> <p>For more information, see Message processing.</p> <p><u>Default value:</u> <code>CanChangeBody</code> = Yes</p>
<pre>ProcessingTimeout = {time}</pre>	<p>Timeout for the <code>drweb-milter</code> module to wait for a message to be scanned.</p> <p>It is recommended to set this parameter value greater than the <code>SendTimeout</code> parameter value from the [MailBase] section.</p> <p>Note that value of the <code>IPTimeout</code> parameter in the [General] section is also considered. Dr.Web MailD selects greater value among values of the <code>ProcessingTimeout</code> and <code>IPTimeout</code> parameters. If during the selected timeout Dr.Web MailD does not return response to <code>drweb-milter</code>, action specified in the <code>ProcessingErrors</code> parameter is performed (see below) and "broken pipe" errors are recorded in the Dr.Web MailD log.</p> <p><u>Default value:</u> <code>ProcessingTimeout</code> = 40s</p>
<pre>ProcessingErrors = {action}</pre>	<p>Action applied to messages that caused scanning errors.</p> <p>Only one of these actions can be specified: tempfail, discard, pass, reject.</p> <p><u>Default value:</u> <code>ProcessingErrors</code> = reject</p>
<pre>MinPersistConnection = {numerical value}</pre>	<p>Minimum number of connections to the <code>drweb-maild</code> module.</p> <p>In the current version, this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required.</p> <p><u>Default value:</u> <code>MinPersistConnection</code> = 2</p>
<pre>UseStat = {logical}</pre>	<p>Statistics of connections to the <code>drweb-maild</code> module.</p> <p>Statistics is logged when <code>drweb-milter</code> process receives <code>SIGUSR1</code> signal.</p> <p><u>Default value:</u> <code>UseStat</code> = No</p>



MaxFreetime = {time}	Maximum idle time before closing all connections with the drweb-maild module. Default value: MaxFreetime = 2m
ReplyPoolOptions = {pool options}	Settings for a pool of threads processing responses from the drweb-maild module. Default value: ReplyPoolOptions = auto

[Qmail] Section

In the [Qmail] section, settings for [interaction](#) between **Dr.Web MailD** and **Qmail** mail transfer system are specified. This section is included in the **Dr.Web MailD** configuration file of a package designed for operation with the MTA mentioned above.

ProcessingTimeout = {time}	Timeout for the drweb-qmail module to wait for a message to be scanned. It is recommended to set this parameter value greater than the SendTimeout parameter value from the [MailBase] section . Default value: ProcessingTimeout = 40s
ReadingTimeout = {time}	Timeout to receive envelope and message body from the qmail-queue module. Default value: ReadingTimeout = 20m
ProcessingErrors = {action}	Action applied to the messages invoked scanning errors. Only one of these actions can be specified: tempfail, discard, pass, reject. Default value: ProcessingErrors = reject
MainPoolOptions = {pool options}	Settings for a pool of threads processing requests. Default value: MainPoolOptions = auto
ReplyPoolOptions = {pool options}	Settings for a pool of threads processing responses from the drweb-maild module. Default value: ReplyPoolOptions = auto
ListenUnixSockets = {address}	List of UNIX sockets for the drweb-qmail module to receive requests from the qmail-queue module for message scan. Sockets in this list must be also specified in the list of files monitored by the qmail-queue module. This list can be viewed with qmail-queue --help command. Default value: ListenUnixSockets = local:%var_dir/ipc/.qmail



QmailQueue = {path to file}	Path to the original initial qmail-queue file.
	<u>Default value:</u> QmailQueue = /var/qmail/bin/qmail-queue.original

[IMAP] Section

In the [IMAP] section, settings for **drweb-imap** module are specified (the **IMAP filter** component used to intercept mail via IMAP protocol when [working with mail clients](#)):

ServerAddress = {address}	Address used by the filter to connect to the IMAP server.
	<u>Default value:</u> ServerAddress = inet:imap@127.0.0.1

ListenAddress = {address list}	A list of socket addresses used to receive requests from clients. The following types of addresses can be specified: <code>inet:..</code> or <code>inet-ssl:..</code> (if you use TLS/SSL encryption). The latter address type requires IMAPS protocol to be used by the IMAP filter.
	<u>Default value:</u> ListenAddress = inet:5200@0.0.0.0

ServerTLSSettings = {TLS/SSL settings}	TLS/SSL settings for server communications over IMAP. It is possible to connect as TSL/SSL server only if a certificate (certificate) and private key (<code>private_key_file</code>) are specified and <code>inet-ssl</code> socket is used. Example: ServerTLSSettings = use_sslv2 no, private_key_file /path/to/pkey, certificate / path/to/certificate Please note that the user whose privileges are used by the IMAP filter (usually, <code>drweb</code> user), must have read access to the file with the certificate. In the current program version caching of SSL sessions is not supported.
	<u>Default value:</u> ServerTLSSettings =

ClientTLSSettings = {TLS/SSL settings}	TLS/SSL settings used for client communications over IMAP. Example: ClientTLSSettings = use_sslv2 no, private_key_file /path/to/pkey, certificate / path/to/certificate Please note that the user whose privileges are used by the IMAP filter (usually, <code>drweb</code> user), must have read access to the file with the certificate. In the current program version caching of SSL sessions is not supported.
	<u>Default value:</u> ClientTLSSettings =

IoTimeout = {time}	Timeout for all input-output operations with the client's socket, when operation is in progress.
	<u>Default value:</u> IoTimeout = 60s



<code>ProcessingTimeout = {time}</code>	<p>Timeout for drweb-maild to process messages.</p> <p>Default value: ProcessingTimeout = 60s</p>
<code>MinFilterToMaidConnections = {numerical value}</code>	<p>Minimum number of connections between IMAP filter and drweb-maild.</p> <p>Default value: MinFilterToMaidConnections = 2</p>
<code>MaxFilterToMaidConnections = {numerical value}</code>	<p>Maximum number of connections between IMAP filter and drweb-maild module.</p> <p>When the value is set to 0, number of connections is not limited.</p> <p>Default value: MaxFilterToMaidConnections = 0</p>
<code>FilterToMaidKeepAliveTime = {time}</code>	<p>Maximum period for retention of inactive connections between IMAP filter and drweb-maild if number of connections exceeds the specified minimum.</p> <p>To interact with drweb-maild, IMAP filter maintains several connections with it, and each connection can handle one operation. If no more connections are available, new connections are created until their amount reaches the maximum allowed (specified in the MaxFilterToMaidConnection parameter value).</p> <p>If connections are inactive during a time period specified in the FilterToMaidKeepAliveTime parameter, they are closed. Total amount of opened connections cannot be less than the value specified in the MinFilterToMaidConnections parameter.</p> <p>Default value: FilterToMaidKeepAliveTime = 60s</p>
<code>CallbackPoolOptions = {pool options}</code>	<p>Settings of an auxiliary thread pool.</p> <p>Threads handle end of message processing signals from the drweb-maild.</p> <p>Default value: CallbackPoolOptions = auto</p>
<code>PoolOptions = {pool seoptions}</code>	<p>Settings of the main thread pool.</p> <p>These threads handle connections from clients. Each connection requires a new thread, otherwise some clients are disconnected while waiting for a free thread.</p> <p>Default value: PoolOptions = auto</p>
<code>MaxConnections = {numerical value}</code>	<p>Maximum number of incoming connections.</p> <p>If 0 is specified as a value of this parameter, the number of incoming connections is not limited.</p> <p>Default value: MaxConnections = 0</p>
<code>MaxConnectionsPerIp = {numerical value}</code>	<p>Maximum number of simultaneous connections from one IP address.</p>



	<p>If 0 is specified as a value of this parameter the number of connections is not limited.</p> <p><u>Default value:</u> MaxConnectionsPerIp = 0</p>
DisablePlainText = {logical}	<p>Do not allow the client to send login and password as plain text (in an unencrypted format).</p> <p>It requires OpenSSL to be configured in advance.</p> <p><u>Default value:</u> DisablePlainText = no</p>
DoS_Blackhole = {logical}	<p>Enables to drop connection without sending an error message to a client if there are too many simultaneous connections from one IP address.</p> <p><u>Default value:</u> DoS_Blackhole = no</p>
MaxCommandLength = {size}	<p>Maximum size of a command for the IMAP protocol.</p> <p>Each command is a string sent from a client to the server. Maximum possible size of this command is about 1000 bytes according to the current RFC.</p> <p>Please note that if this parameter value is small (less than 10 bytes), clients' commands are not processed.</p> <p><u>Default value:</u> MaxCommandLength = 1000b</p>
MaxCachedHeadersPerMail = {size}	<p>Maximum amount of memory to be allocated to store frequently used headers.</p> <p>IMAP filter caches main message headers in random access memory to speed up access to them.</p> <p>If 0 is specified as a value of this parameter, amount of allocated memory is not controlled (memory size is restricted only to the size of available memory).</p> <p><u>Default value:</u> MaxCachedHeadersPerMail = 64k</p>
MaxLettersPerUser = {numerical value}	<p>Maximum number of messages to be cached during one session.</p> <p>IMAP filter maintains cache of checked messages because IMAP protocol allows the client to perform lots of partial requests to one message.</p> <p>In most cases requests are performed sequentially, but if a user accesses several records, it is necessary to cache more than one message.</p> <p>If the value of this parameter is set to 0 (which is strongly NOT recommended), number of cached messages is unlimited.</p> <p><u>Default value:</u> MaxLettersPerUser = 6</p>
MaxDiskPerUser = {size}	<p>Maximum amount of disk space to be occupied by cached messages.</p> <p><u>Default value:</u> MaxDiskPerUser = 10m</p>



OnFilterErrors = {actions}	Action to be applied to the message, when an error occurs before the message is transmitted to the drweb-maild module. Possible values are <code>reject</code> or <code>pass</code> . <u>Default value:</u> OnFilterErrors = <code>reject</code>
--------------------------------------	---

[POP3] Section

In the [POP3] section, settings for **drweb-pop3** module are specified (the **POP3 filter** component used to intercept mail via POP3 protocol when [working with mail clients](#)):

ServerAddress = {address}	Address used by the POP3 filter to connect to the POP3 server. <u>Default value:</u> ServerAddress = <code>inet:pop3@localhost</code>
ListenAddress = {address list}	A list of socket addresses used to receive requests from clients. The following types of addresses can be specified: <code>inet:..</code> or <code>inet-ssl:..</code> (if you use TLS/SSL encryption). The latter address type requires POP3S protocol to be used by the POP3 filter. <u>Default value:</u> ListenAddress = <code>inet:5110@localhost</code>
ServerTLSSettings = {TLS/SSL settings}	TLS/SSL settings for server communications over POP3. Settings are separated by commas. It is possible to connect as TLS/SSL server only if a certificate and private key (<code>private_key_file</code>) are specified and <code>inet-ssl</code> socket is used. Example: ServerTLSSettings = <code>use_sslv2 no, private_key_file /path/to/pkey, certificate /path/to/certificate</code> Please note that the user whose privileges are used by the POP3 filter (usually, <code>drweb</code> user), must have read access to the file with the certificate. <u>Default value:</u> ServerTLSSettings =
ClientTLSSettings = {TLS/SSL settings}	TLS/SSL settings for client communications over POP3. Settings are separated by commas. Example: ClientTLSSettings = <code>use_sslv2 no, private_key_file /path/to/pkey, certificate /path/to/certificate</code> Please note that the user whose privileges are used by the POP3 filter (usually, <code>drweb</code> user), must have read access to the file with the certificate. <u>Default value:</u> ClientTLSSettings =
IoTimeout = {time}	Timeout for all input-output operations with a client socket when the operation is in progress. <u>Default value:</u> IoTimeout = <code>60s</code>



<code>ProcessingTimeout = {time}</code>	<p>Timeout for drweb-maild to process messages.</p> <p>Default value: ProcessingTimeout = 60s</p>
<code>MinFilterToMaidConnections = {numerical value}</code>	<p>Minimum number of connections between the POP3 filter and drweb-maild.</p> <p>Default value: MinFilterToMaidConnections = 2</p>
<code>MaxFilterToMaidConnections = numerical value}</code>	<p>Maximum number of connections between the POP3 filter and drweb-maild module.</p> <p>When the value is set to 0, number of connections is not limited.</p> <p>Default value: MaxFilterToMaidConnections = 0</p>
<code>FilterToMaidKeepAliveTime = {time}</code>	<p>Maximum period for retention of inactive connections between POP3 filter and drweb-maild if number of connections exceeds the specified minimum.</p> <p>To interact with drweb-maild, POP3 filter maintains several connections with it, and each connection can handle one operation. If no more connections are available, new connections are created until their amount reaches the maximum allowed (specified in the MaxFilterToMaidConnection parameter value).</p> <p>If connections are inactive during a time period specified in the FilterToMaidKeepAliveTime parameter, the connections are closed. At that, total amount of opened connections cannot be less than the value specified in the MinFilterToMaidConnections parameter.</p> <p>Default value: FilterToMaidKeepAliveTime = 30s</p>
<code>PoolOptions = {pool options}</code>	<p>Settings of the main thread pool.</p> <p>These threads handle connections from clients.</p> <p>Each connection requires a new thread, otherwise some clients are disconnected while waiting for a free thread.</p> <p>Default value: PoolOptions = auto</p>
<code>CallbackPoolOptions = {pool options}</code>	<p>Settings of an auxiliary thread pool.</p> <p>Threads handle end of message processing signals from drweb-maild.</p> <p>Default value: CallbackPoolOptions = auto</p>
<code>MaxConnections = {numerical value}</code>	<p>Maximum number of incoming connections.</p> <p>If 0 is specified as a value of this parameter the number of incoming connections is not limited.</p> <p>Default value: MaxConnections = 0</p>
<code>DoS_Blackhole = {logical}</code>	<p>Enables to drop connection without sending an error message to client if there are too many simultaneous connections from one IP</p>



	address. <u>Default value:</u> DoS_Blackhole = no
DisablePlainText = {logical}	Do not allow the client to send login and password as plain text (in an unencrypted format). It requires OpenSSL to be configured in advance. <u>Default value:</u> DisablePlainText = no
MaxConnectionsPerIp = {numerical value}	Maximum number of simultaneous connections from one IP address. If 0 is specified as a value of this parameter the number of incoming connections is not limited. <u>Default value:</u> MaxConnectionsPerIp = 0
MaxCommandLength = {size}	Maximum size of a command for the POP3 protocol. Each command is a string, which is sent from the client to the server. Maximum possible size of this command is about 1000 bytes according to the current RFC. Please note that if this parameter value is small (less than 10 bytes), clients' commands are not processed. <u>Default value:</u> MaxCommandLength = 1000b
OnFilterErrors = {action}	Action to be applied to the message, when an error occurs before the message is transmitted to the drweb-maild module. Possible values are <code>reject</code> or <code>pass</code> . <u>Default value:</u> OnFilterErrors = <code>reject</code>

Data Sources

This chapter provides you with description of sections that contain parameters of connection to data sources used in `Lookup` and `Storage`. As a data source, it is possible to use LDAP, databases, and text files with regular structure.

Note that these sections contain only general parameters used in `Lookup` and `Storage` by default. Some parameters (as marked in their description) can be locally overridden in each concrete `Lookup` (`Storage`).

Dr.Web MailD configuration file can contain any sections with settings of connection to data sources. Those sections are actual that are used in `Lookup` and `Storage`. Sections that contain parameters of connections to unused data sources do not influence **Dr.Web for UNIX mail servers** operation.

[LDAP] Section

In the `[LDAP]` section, settings to establish and maintain interaction between **Dr.Web MailD** and LDAP server are specified:

Lib = {path to file}	Path to OpenLDAP library version 2.0 or later. Library must be built with thread support (that is, "_r" suffix must
-----------------------------	--



	be present in the file name). Library is located using <code>dlopen</code> system call (please refer to the corresponding documentation). In the current version this parameter cannot be changed with <code>SIGHUP</code> signal, restart of Dr.Web MailD is required.
	<u>Default value:</u> Lib = /usr/lib/libldap_r.so



Please note that when using `libldap_r.so` library in **FreeBSD** 6.4/amd64 the following error may occur:

Undefined symbol "gethostbyname_r"

Hostname = {string}	LDAP server hostname. If the parameter value is not specified, <code>localhost</code> is used. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> Hostname =
Port = {numerical value}	LDAP server port. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> Port = 389
Timeout = {time}	Timeout for LDAP requests. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> Timeout = 10s
Version = {string}	LDAP protocol version. To enable secure data transfer with TLS/SSL, use LDAP protocol version 3 or later. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> Version = 3
Bind = {logical}	Enables binding before making requests. For LDAP protocol version 3, binding is not necessary. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> Bind = No
BindDn = {string}	Unique name for binding. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> BindDn =
BindPw = {string}	Password used for binding. The parameter value can also be locally overridden in a Lookup . <u>Default value:</u> BindPw =



SearchBase = {string}	Base DN to start search from (RFC2253). Default value: SearchBase =
SizeLimit = {numerical value}	Maximum number of strings received in response to the single database request. When parameter value is set to 0, maximum number of received strings is not limited. The parameter value can also be locally overridden in a Lookup . Default value: SizeLimit = 0
Dereference = {3 2 1 0}	Permissions for LDAP aliases: <ul style="list-style-type: none">• 0 – never;• 1 – when searching;• 2 – when locating base object for search;• 3 – always. The parameter value can also be locally overridden in a Lookup . Default value: Dereference = 0
ChaseReferrals = {numerical value}	LDAP_OPT_REFERRALS setting. To set this parameter, LDAP protocol version 3 or later is required. The parameter value can also be locally overridden in a Lookup . Default value: ChaseReferrals = 0
SkipDomains = {LookupLite}	List of domains for which request to database is not required. This parameter often helps improve performance and considerably reduce server load. Please note that the parameter value is LookupLite . The parameter value can also be locally overridden in a Lookup . Default value: SkipDomains =
OnError = {ignore exception}	Sets mode of error handling (errors that occur when connecting to the specified data source). Allowed modes: <ul style="list-style-type: none">• <code>ignore</code> – ignore the error and continue message processing (the error is only recorded in log);• <code>exception</code> – throw an exception which will be handled as an error of message processing. The handling method corresponds to the value of the ProcessingError parameter specified for the module that was processing the message when the error occurred. The parameter value can also be locally overridden in Lookup . Default value: OnError = ignore
CheckPeriod = {time}	Maximum idle time for LDAP connection to be closed.



	Check for LDAP inactive connections is performed using the same time period.
	Default value: CheckPeriod = 2m

Dr.Web MailD uses **OpenLDAP** library for connection to LDAP (library version must be 2.0 and later).

If the specified LDAP-server is not available, **Dr.Web MailD** attempts to establish connection until the timeout occurs (specified in the **Timeout** parameter). When the time expires, an error is logged and processed in accordance with the **OnError** parameter value.

[Oracle] Section

In the [Oracle] section, settings for interaction between **Dr.Web MailD** and **Oracle** database are specified:

Lib = {path to file}	<p>Path to library that supports Oracle OTL version 8 or later.</p> <p>Library must be built with thread support. The library is searched according to the standard rules used by <code>dlopen</code> system call (please refer to the corresponding documentation).</p> <p>In the current version this parameter cannot be changed with SIGHUP signal, restart of Dr.Web MailD is required.</p> <p>Default value: Lib =</p>
ConnectData = {string}	<p>Oracle connection parameters.</p> <p>The following two formats are supported:</p> <ul style="list-style-type: none"> "USER/PASSWORD@CONNECTION" - Oracle syntax; "DSN=value;UID=value;PWD=value" - ODBC syntax. <p>To start working with Oracle, specify DSN that references to the required database.</p> <p>To see the rules of setting ConnectData value with the use of Oracle syntax, refer to the notes under the table.</p> <p>In addition, it is recommended to use the <code>connect_timeout</code> parameter in the ConnectData string (this parameter specifies connection timeout).</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p>Default value: ConnectData =</p>
SizeLimit = {numerical value}	<p>Maximum number of strings received in response to a single database query.</p> <p>When the parameter value is set to 0, maximum number of received strings is not limited.</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p>Default value: SizeLimit = 0</p>
SkipDomains = {LookupLite}	<p>List of domains for which query to database is not required.</p> <p>This parameter often allows to improve total performance and considerably reduce server load.</p> <p>Please note that the parameter value is LookupLite.</p>



	The parameter value can also be locally overridden in a Lookup .
	Default value: SkipDomains =
OnError = {ignore exception}	Sets the mode of error handling (errors that occur in <code>Lookup</code> when connecting to data source). Allowed modes: <ul style="list-style-type: none"> <code>ignore</code> – ignore the error and continue message processing (the error is only logged); <code>exception</code> – generate an exception which is handled as a message processing error. The handling method corresponds to the ProcessingError parameter value set for the module in operation of which this error occurred.
	The parameter value can also be locally overridden in Lookup .
	Default value: OnError = ignore

Notes:

1. **Dr.Web MailD** uses the `libc1ntsh` library to connect to **Oracle** databases (it is distributed with **Oracle client** and supports OTL v8 or later).
2. To connect to **Oracle** database, specify username, password and connection name for the **ConnectData** parameter in the following format `user/password@CONNECTION`.

You can specify the connection name in one of the following ways:

- if **Oracle** DBMS is installed on the same computer as **Dr.Web MailD**, define environment variable `ORACLE_HOME` for **Dr.Web MailD** (as per the **Oracle** DBMS documentation). After that, specify one of the TNS names in file `$ORACLE_HOME/network/admin/tnsnames.ora` as the connection name;
- copy the connection description (without line breaks) from the `$ORACLE_HOME/network/admin/tnsnames.ora` file located on the server.

Example:

Let us assume that there is an `tnsnames.ora` file:

```
CONNECTIONNAME =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CONNECTIONNAME)
    )
  )
```

As the connection string (the **ConnectData** parameter value), it is possible to specify either:

```
user/password@CONNECTIONNAME
```

or:

```
user/pasword@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST =
localhost) (PORT = 1521)) (CONNECT_DATA = SERVER = DEDICATED)
(SERVICE_NAME = CONNECTIONNAME))
```

3. If the specified host or database are not accessible, connection is attempted to be established until timeout occurs (the timeout value is specified for the `connect_timeout` parameter in the connection string). After that, an error is fixed and handled according to the action set for the **OnError** parameter.



[ODBC] Section

In the [ODBC] section, settings for interaction between **Dr.Web MailD** and databases via **ODBC** are specified:

<code>Lib = {path to file}</code>	<p>Path to the library that supports ODBC version 3.0 or later.</p> <p>Library must be built with thread support. UnixODBC is recommended. The library is searched according to the standard rules used by <code>dlopen</code> system call (please refer to the corresponding <code>dlopen</code> documentation).</p> <p>In the current version this parameter cannot be changed with SIGHUP signal, restart of Dr.Web MailD is required.</p> <p><u>Default value:</u> Lib = <code>/usr/lib/libodbc.so</code></p>
<code>ConnectData = {string}</code>	<p>ODBC connection parameters.</p> <p>The following two formats are supported:</p> <ul style="list-style-type: none">• <code>USER/PASSWORD/@DSN</code> - Oracle syntax;• <code>DSN=value;UID=value;PWD=value</code> - ODBC syntax. <p>To start working with ODBC, specify at least DSN.</p> <p>In addition, it is recommended to use the <code>connect_timeout</code> parameter in ConnectData string (this parameter specifies wait time for the connection to be established).</p> <p>This parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> ConnectData =</p>
<code>SizeLimit = {string}</code>	<p>Maximum number of strings received in response to a single database request.</p> <p>When parameter value is set to 0, maximum number of received strings is not limited.</p> <p>This parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> SizeLimit = 0</p>
<code>SkipDomains = {LookupLite}</code>	<p>List of domains for which query to database is not required.</p> <p>This parameter often allows to improve total performance and considerably reduce server load.</p> <p>Please note that the parameter value is LookupLite.</p> <p>This parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> SkipDomains =</p>
<code>OnError = {ignore exception}</code>	<p>Sets the mode of error handling (errors that occur in <code>Lookup</code> when connecting to the specified data source).</p> <p>Allowed modes:</p> <ul style="list-style-type: none">• <code>ignore</code> - ignore the error and continue message processing (the error is only logged);• <code>exception</code> - generate an exception which is handled as a message processing error. The handling method corresponds to the ProcessingError parameter value set for the module in operation of which this error occurred.



	The parameter value can also be locally overridden in a Lookup .
	Default value: OnError = ignore

To connect to **ODBC** data sources, **Dr.Web MailD** uses any library which supports **ODBC** version 3.0 or later. The used library must be built with thread support. It is recommended to use **UnixODBC** library version 2.0 or later.

If the specified host or database are not accessible, connection is attempted to be established until timeout occurs (the timeout value is specified for the `connect_timeout` parameter in the connection string). After that, an error is fixed and handled according to the action set for the `OnError` parameter.

[SQLite] Section

In the [SQLite] section, settings for interaction between **Dr.Web MailD** and **SQLite** database are specified:

Database = {path to file}	Path to the SQLite database file. Default value: Database =
SizeLimit = {integer}	Maximum number of strings received in response to a single database request. When the parameter value is set to 0, maximum number of received strings is not limited. Default value: SizeLimit = 1
Lib = {path to file}	Path to the <code>libsqlite3.so</code> library. Default value: Lib = <code>/usr/lib/libsqlite3.so</code>
BusyTimeout = {time in milliseconds}	Timeout for Dr.Web MailD (in milliseconds) to make an entry to the database. Default value: BusyTimeout = 2000
SkipDomains = {LookupLite}	List of the domains for which query to the database is not required. This parameter often allows to improve total performance and considerably reduce server load. Please note that the parameter value is LookupLite . The parameter value can also be locally overridden in a Lookup . Default value: SkipDomains =
OnError = {ignore exception}	Sets the mode of error handling (errors that occur in <code>Lookup</code> when connecting to the specified data source). Allowed modes: <ul style="list-style-type: none"> <code>ignore</code> – ignore the error and continue message processing (the error is only logged); <code>exception</code> – generate an exception which is handled as a



	<p>message processing error. The handling method corresponds to the ProcessingError parameter value set for the module in operation of which this error occurred.</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p>Default value: OnError = ignore</p>
--	---

Note the following features of using **SQLite** DBMS:

- Every time when information is written to the database, its file is locked. So, if more than one program is working with the same **SQLite** database file, it is possible that a writing process cannot obtain exclusive access to the database during the time specified in the **BusyTimeout** parameter. As a result, the process is aborted with the following error: "Database is locked".
- Avoid using GUI for **SQLite**, as graphical interface can lock the databases for future use.
- If an external process locked the database for a long time, statistics export errors can occur. A conflict may also occur if **Dr.Web MailD** is configured to export different types of statistics into the same **SQLite** database file and the specified time is too small.
- If the **SQLite** database, that provides settings for several parameters (using [Lookup](#) of `sqlite` type) was not accessible over a period of time and then connection to the database was reestablished, it is necessary to send a `SIGHUP` signal to `drweb-maild` module in order to reconnect **Dr.Web MailD** and **SQLite**.

[Firebird] Section

In the [Firebird] section, settings for interaction between **Dr.Web MailD** and **Firebird** database are specified:

Host = {hostname}	<p>Firebird database hostname.</p> <p>Default value: Host = localhost</p>
Database = {string}	<p>Name of the Firebird database.</p> <p>Default value: Database =</p>
User = {string}	<p>Firebird database user name.</p> <p>Default value: User =</p>
Password = {string}	<p>Firebird database password.</p> <p>Default value: Password =</p>
Charset = {string}	<p>Firebird character set.</p> <p>Default value: Charset = us-ascii</p>
SizeLimit = {integer}	<p>Maximum number of strings received in response to a single database request.</p> <p>When the parameter value is set to 0, maximum number of received strings is not limited.</p> <p>Default value: SizeLimit = 10</p>



Lib = {path to file}	Path to the libFBclient.so library. Default value: Lib = /usr/lib/libFBclient.so
SkipDomains = {LookupLite}	List of domains for which request to the database is not required. This parameter often allows to improve total performance and considerably reduce server load. Please note that the parameter value is LookupLite. The parameter value can also be locally overridden in a Lookup . Default value: SkipDomains =
OnError = {ignore exception}	Sets the mode of error handling (errors that occur in <code>Lookup</code> when connecting to the specified data source). Allowed modes: <ul style="list-style-type: none">• <code>ignore</code> – ignore the error and continue message processing (the error is only logged);• <code>exception</code> – generate an exception which is handled as a message processing error. The handling method corresponds to the ProcessingError parameter value set for the module in operation of which this error occurred. The parameter value can also be locally overridden in a Lookup . Default value: OnError = ignore

[PostgreSQL] Section

In the [PostgreSQL] section settings for interaction between **Dr.Web MailD** and **PostgreSQL** database are specified:

ConnectionsString = {string}	String with settings for connection to PostgreSQL database. The string can be empty (in this case, default parameters are used) or it can contain one or more parameter settings separated by white space. Parameters are specified in the following format: <code>keyword = value</code> . Spaces around the equal sign are optional. To specify an empty value or a value containing spaces, enclose it in single quotes. If the string is empty, default parameters are used. For details, refer to http://www.postgresql.org/docs/9.3/static/libpq-connect.html . In addition is recommended to use the <code>connect_timeout</code> parameter that specifies wait time for a connection to be established. Examples: ConnectionString = host=localhost port=5432 user=ai password=qwerty dbname=drweb ConnectionString = hostaddr=127.0.0.1 port=5432 dbname=mailddb user=mailddbuser password=Str0ngPaSSw0rd connect_timeout=5s Default value: ConnectionString =
-------------------------------------	---



SizeLimit = {integer}	<p>Maximum number of strings received in response to a single database request.</p> <p>When the parameter value is set to 0, maximum number of received strings is not limited.</p> <p><u>Default value:</u> SizeLimit = 10</p>
Lib = {path to file}	<p>Path to libpq.so library.</p> <p><u>Default value:</u> Lib = /usr/lib/libpq.so</p>
SkipDomains = {LookupLite}	<p>List of domains for which request to database is not required.</p> <p>This parameter often allows to improve total performance and considerably reduce server load.</p> <p>Please note that the parameter value is LookupLite.</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> SkipDomains =</p>
OnError = {ignore exception}	<p>Sets the mode of error handling (errors that occur in <code>Lookup</code> when connecting to the specified data source).</p> <p>Allowed modes:</p> <ul style="list-style-type: none"> • <code>ignore</code> – ignore the error and continue message processing (the error is only logged); • <code>exception</code> – generate an exception which is handled as a message processing error. The handling method corresponds to the ProcessingError parameter value set for the module in operation of which this error occurred. <p>The parameter value can also be locally overridden in a Lookup value.</p> <p><u>Default value:</u> OnError = ignore</p>

If the specified host or database are not accessible, connection is attempted to be established until timeout occurs (the timeout value is specified for the `connect_timeout` parameter in the connection string; in the example above the timeout is set to 5 seconds). After that, an error is fixed and handled according to the action set for the `OnError` parameter.

[MySQL] Section

In the [MySQL] section settings for interaction between **Dr.Web MailD** and **MySQL** database are specified:

User = {string}	<p>MySQL database user name.</p> <p><u>Default value:</u> User =</p>
Password = {string}	<p>MySQL database password.</p> <p><u>Default value:</u> Password =</p>
DB =	<p>Name of the MySQL database.</p>



	<p><u>Default value:</u> DB =</p>
Host = {hostname}	<p>Name of the host used by MySQL database.</p> <p><u>Default value:</u> Host = localhost</p>
Port = {address}	<p>Port used to connect to MySQL database.</p> <p>It is also required to specify a prefix that indicates the socket type: TCP or UNIX.</p> <p>Example:</p> <p>When TCP-socket is used: Port = tcp://1234</p> <p>When UNIX socket is used: Port = unix:///path/to/socket</p> <p><u>Default value:</u> Port =</p>
Connections = {numerical value}	<p>Number of simultaneous connections to the MySQL database.</p> <p>When the parameter value is set to 0, connections are created as a query to the database is performed (it usually takes additional time). Connections opened in advance can service database queries in turn without consuming time for reconnection.</p> <p><u>Default value:</u> Connections = 4</p>
SizeLimit = {numerical value}	<p>Maximum number of strings received in response to a single database request.</p> <p>When the parameter value is set to 0, maximum number of received strings is not limited.</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> SizeLimit = 10</p>
Lib = {path to file}	<p>Path to libmysqlclient.so library.</p> <p>Library must be built with thread support.</p> <p><u>Default value:</u> Lib = /usr/lib/libmysqlclient_r.so</p>
SkipDomains = {LookupLite}	<p>List of domains for which request to database is not required.</p> <p>This parameter often allows to improve total performance and considerably reduce server load.</p> <p>Please note that the parameter value is LookupLite.</p> <p>The parameter value can also be locally overridden in a Lookup.</p> <p><u>Default value:</u> SkipDomains =</p>



```
OnError =
{ignore | exception}
```

Sets the mode of error handling (errors that occur in `Lookup` when connecting to the specified data source).

Allowed modes:

- `ignore` – ignore the error and continue message processing (the error is only logged);
- `exception` – generate an exception which is handled as a message processing error. The handling method corresponds to the `ProcessingError` parameter value set for the module in operation of which this error occurred.

The parameter value can also be locally overridden in a [Lookup](#).

Default value:

```
OnError = ignore
```



Note that when using `libmysqlclient_r.so` library in **FreeBSD 6.4/amd64** the following error may occur:

```
Undefined symbol "gethostbyname_r"
```

If the specified host or **MySQL** database are not accessible, connection is attempted to be established during 2 seconds. After the timeout occurs, an error is logged and handled according to the `OnError` parameter value.

[CDB] Section

In the [CDB] section settings for interaction between **Dr.Web MailD** and **CDB** database are specified:

```
Sources =
{path to file}
```

Path to the **CDB** database file.

Default value:

```
Sources =
```

```
SkipDomains =
{LookupLite}
```

List of domains for which request to database is not required.

This parameter often allows to improve total performance and considerably reduce server load.

Please note that the parameter value is [LookupLite](#).

The parameter value can also be locally overridden in a [Lookup](#).

Default value:

```
SkipDomains =
```

```
OnError =
{ignore | exception}
```

Sets the mode of error handling (errors that occur in `Lookup` when connecting to the specified data source).

Allowed modes:

- `ignore` – ignore the error and continue message processing (the error is only logged);
- `exception` – generate an exception which is handled as a message processing error. The handling method corresponds to the `ProcessingError` parameter value set for the module in operation of which this error occurred.

The parameter value can also be locally overridden in a [Lookup](#).

Default value:

```
OnError = ignore
```



CDB database is a read-only storage of [alphanumeric_key]:[alphanumeric_value] pairs.

You may use **tinycdb** package to create a database file. Each file is represented as a single table; name of the table is the name of the corresponding file (without the full path to it: /path/to/table.cdb -> table.cdb).

CDB database does not support SQL query language, so the driver emulates the single SQL command to unify operation with **Lookup**:

```
select * from @tablename where key='@string'
```

where @tablename is a filename.

[Berkeley] Section

In the [Berkeley] section settings for interaction between **Dr.Web MailD** and **Berkeley** database are specified:

Databases = {path to file}	Path to the Berkeley database file Default value: Databases =
Environment = {path to directory}	Path to the directory where Berkeley database lock files are stored. Default value: Environment =
SizeLimit = {integer}	Maximum number of bytes received in response to a single database request. Values must be within the following range: from 1024 to 65536. Other values are converted to the closest valid value. Default value: SizeLimit = 1
Lib = {path to file}	Path to the libdb.so library. During installation, /usr/lib/libdb.so symbolic link to the library is usually created. If not, it is necessary to specify the right library version (that is, /usr/lib/libdb-4.5.so)/ Default value: Lib = /usr/lib/libdb.so
SkipDomains = {LookupLite}	List of domains for which query to the database is not required. This parameter often allows to improve total performance and considerably reduce server load. Please note that the parameter value is LookupLite . The parameter value can also be locally overridden in a Lookup . Default value: SkipDomains =
OnError = {ignore exception}	Sets the mode of error handling (errors that occur in Lookup when connecting to the specified data source).



Allowed modes:
<ul style="list-style-type: none">• <code>ignore</code> – ignore the error and continue message processing (the error is only logged);• <code>exception</code> – generate an exception which is handled as a message processing error. The handling method corresponds to the <code>ProcessingError</code> parameter value set for the module in operation of which this error occurred.
The parameter value can also be locally overridden in a Lookup .
Default value:
<code>OnError</code> = <code>ignore</code>

Dr.Web MailD uses a library version 4.3 – 4.6.

Internal Proxy

This chapter provides you with description of sections which contain parameters of **Dr.Web for UNIX mail servers** proxying.

[Proxying](#) is used for transparent distribution of **Dr.Web MailD modules** by several hosts.

Please note that if proxying is used, **Sender-Receiver** component pairs operate on some hosts and **MailD core** core processing component - on the others. On hosts where **Sender** and **Receiver** components work, **Proxy client** component is run; on hosts with **MailD core - Proxy server** component.

On the host, where **Proxy server** operates, **Dr.Web MailD** configuration file must have the `[ProxyServer]` [section](#), and on the host where **Proxy client** component operates - `[ProxyClient]` [section](#).

If proxying is not used, both sections in the configuration file can be absent.

[ProxyClient] Section

In the `[ProxyClient]` section settings for `drweb-proxy-client` module (**Proxy client** component) are specified:

<code>ProxyServersAddresses</code> = {list of addresses}	List of socket addresses used by <code>drweb-proxy-server</code> components. Addresses are specified as follows: <code>ADDRESS1 [WEIGHT1], ADDRESS2 [WEIGHT2] ..</code> where <code>ADDRESS</code> has a basic address type, and <code>WEIGHT</code> is an optional numeric value in the range 0-100, defining a "weight" of this address. The <code>WEIGHT</code> value determines a relative work load on a certain host in the network. The greater the value, the greater the load on the server. Mail received from the Receiver component working on the same host as <code>drweb-proxy-client</code> , is transmitted for scanning to the corresponding socket addresses. There must be at least one valid server address specified. Addresses are selected according to the algorithm described in Using Internal Proxy .
	Default value: <code>ProxyServersAddresses</code> = <code>inet:8088@SERVER-IP</code>



<pre>Address = {list of addresses}</pre>	<p>List of socket addresses used by Sender to receive send requests from drweb-proxy-server components.</p> <p>drweb-proxy-server components send mail to these addresses according to the value set for the ProxyClientsAddresses parameter from the [ProxyServer] section.</p> <p>Default value: Address = inet:8066@0.0.0.0</p>
<pre>MailPoolOptions = {pool options}</pre>	<p>Settings for a pool of threads processing requests from the Receiver component.</p> <p>Thread pool processes requests from the Receiver component and sends messages to remote drweb-proxy-server components for check. According to the check results, the message is either returned to Receiver or sent via the Sender component.</p> <p>Default value: MailPoolOptions = auto</p>
<pre>SenderPoolOptions = {pool options}</pre>	<p>Settings for a pool of threads processing requests from drweb-proxy-server components to send mail via the Sender component.</p> <p>Before the message is transmitted to Sender, a temporary directory is created, where this message is stored. Sender returns operation results to the drweb-proxy-server.</p> <p>Default value: SenderPoolOptions = auto</p>

[ProxyServer] Section

In the [ProxyServer] section, settings of the **drweb-proxy-server** module (**Proxy-server** component) are specified:

<pre>Address = {list of addresses}</pre>	<p>List of socket addresses used by the drweb-proxy-server component to receive requests from drweb-proxy-client components.</p> <p>drweb-proxy-client transmits messages for checking to the drweb-proxy-server component according to the value set for the ProxyServersAddresses parameter from the [ProxyClient] section.</p> <p>Default value: Address = inet:8088@0.0.0.0</p>
<pre>ProxyClientsAddresses = {list of addresses}</pre>	<p>List of socket addresses used by drweb-proxy-client components to receive send requests.</p> <p>Addresses are specified as follows: ADDRESS1 [WEIGHT1], ADDRESS2 [WEIGHT2] ..</p> <p>where ADDRESS has a basic address type, and WEIGHT is an optional numeric value in the range 0-100, defining a "weight" of this address. This WEIGHT determines a relative work load on a certain host in the network. The greater the value, the greater the load on the server.</p> <p>Socket addresses specified in the parameter value must match socket addresses set as a value of the Address parameter from the [ProxyClient] section.</p>



	<p>Default value: ProxyClientsAddresses = inet:8066@CLIENT-IP</p>
<pre>ReceiverPoolOptions = {pool options}</pre>	<p>Settings for a pool of threads transmitting messages for checking to drweb-maild.</p> <p>Threads in the pool receive check requests from drweb-proxy-client, create a unique ID for this message and pass the message to the drweb-maild for checking.</p> <p>After the check is finished, drweb-proxy-client receives the original or modified message.</p> <p>Default value: ReceiverPoolOptions = auto</p>
<pre>SenderPoolOptions = {pool options}</pre>	<p>Settings for a pool of threads transmitting messages to drweb-proxy-client to send them via the Sender component.</p> <p>Threads in a pool accept requests sending messages from various components, and then transmit these requests for processing to drweb-proxy-client.</p> <p>Processing results are returned to components which requested sending of a message.</p> <p>Default value: SenderPoolSettings = auto</p>

Statistics

During **Dr.Web MailD** operation, statistics of the following two types can be gathered:

- General statistics.
- Statistics on blocked messages.

General statistics contains information on general performance of the **Dr.Web** software for the specified period: number and size of checked messages, number of messages detected as spam and so on. Statistics on blocked messages contains information on certain messages where malicious content was detected.

All statistics is saved to the internal database of **Dr.Web for UNIX mail servers**. General statistics is collected in the internal cache and saved to the database every 5 minutes. Statistics on blocked messages is saved directly to the database and can be [exported](#) if necessary.

Log verbosity levels can be set in the **Detail** parameter from the `[Stat]` [section](#). You can set one of the following values:

- `off` value disables statistics gathering, which allows to increase **Dr.Web for UNIX mail servers** performance. As a result, export of statistics and sending reports are of no use.
- `low` value enables gathering of statistics on operation of the whole suite. As a result, it is possible to export statistics and send reports.
- `medium` value allows gathering statistics on groups, for which the function is not disabled in their settings. Access to group statistics can be gained either via the [control socket](#) or via the web interface.
- `high` value allows gathering statistics on all users listed in the internal database, except for those who disabled this option in their settings. Access to user statistics can be gained either via the [control socket](#) or via the web interface.



Exporting Statistics

Statistics can be exported not only via the **Dr.Web Agent component** as well as by means of **Dr.Web MailD**, via `Storage type`. Both these options can be enabled simultaneously.

Please note that export of statistics via **Dr.Web Agent** is disabled by default.

When all statistics is exported via **Dr.Web Agent**, it either sends the data to **Doctor Web** statistics server (see the `StatisticServerHost`, `StatisticServerPort` and `UUID` parameters in the `[StandaloneMode]` [section](#) of **Dr.Web Agent** configuration file - `%etc_dir/agent.conf`), or sends it to the **Dr.Web** central protection server (corresponding settings can be found in the `[EnterpriseMode]` [section](#) of **Dr.Web Agent** configuration file `agent.conf`).

To enable statistics export via the `Storage type` at first, set `Yes` as an the `ExportStat` parameter value in the `[Stat]` [section](#), then set a value for, at least, one of the following parameters in the `[Stat]` [section](#) and set the commands for the statistics export:

- `ExportBlockObjectsStorage` - list of objects to export statistics on blocked messages;
- `ExportStatStorage` - export of statistics on all messages, processed by **Dr.Web for UNIX mail servers**;
- `ExportPluginStatStorage` - export of statistics on messages processed by each plug-in.

For detailed description of each parameter specified above, refer to the [\[Stat\] Section](#).

Please note that when statistics is exported via **Dr.Web Agent**, two types of statistics are formed and sent separately:

- **Statistics on all detected and processed threats** (viruses and other malicious objects);
- **Statistics on processed messages** and operations on them.

These two types of statistics are gathered separately. So, if an email message which contains 5 different infected objects was processed, statistics of the first type has 5 records (about [actions](#) applied to each object, for example - `cure`) and statistics of the second type has only one record (about an [action](#) applied to the whole message, for example - `pass`).

Quarantine

Mail messages are moved to **Quarantine** on request from any plug-in or `drweb-maild` module. They are saved to the `/quarantine/path/def/name/` directory, where `name` is the name of the module which sent the request.

When a message is moved to **Quarantine**, two files are created:

- The first file is named `name`. Its name is formed according to the `FileNamesMode` and `FileNamesPrefix` parameter values. The file contains the original message body (all `"_"` symbols are replaced with `."` symbols);
- The second file is named `name.envelope` and it contains the original message envelope in the following format:
 - `int4_t` - length of the sender's address;
 - `sN` - sender address;
 - `int4_t` - number of recipients;
 - `int4_t sN` - for each recipient, where `int4_t` is a 4-byte integer with a sign in network byte order.



If the value of the **MoveAll** parameter is set to *Yes*, all messages processed by **Dr.Web MailD** is saved to `/path/def/backup/` directory.

Besides saving the message body to the **Quarantine** directory, the message is registered in the internal database, and the following information is saved to it: message envelope, save time, reason for move and so on.

Quarantine can be effectively managed via the [control socket](#). You can search, send, redirect, remove and perform other actions with the **Quarantine** content.

Maximum period of time for messages to be stored in **Quarantine** can be set with the **StoredTime** parameter. You can also limit the size of **Quarantine** (**MaxSize** parameter) and the number of messages in it (**MaxNumber** parameter).

If several restrictions are specified, all of them are applied simultaneously. **MaxSize** and **MaxNumber** restrictions are checked each time a new message is saved to **Quarantine**. **StoredTime** restriction is checked at certain intervals specified in the **PulseTime** parameter.

drweb-qp utility deletes old messages from **Quarantine** and moves them to the external **DBI** database. It works with **Perl** version 5.0 or later. Path to **drweb-qp** must be specified in the **PathToDrwebQp** parameter. The **drweb-qp** utility is started at certain intervals specified in the **PulseTime** parameter. If the **PulseTime** parameter value is set to 0 and usage of [conttol messages](#) is disabled, the utility does not start.

Using DBI

It is possible to store quarantined messages not only in the file system, but also in a **DBI** storage. To use this feature, **DBI** storage must be pre-configured. For detailed description of **DBI** modules setup and adjustment for operation with databases, refer to documentation on **DBI**. To enable successful transfer of messages to the database, the database must be created with the use of `SQL-ASCII` symbol set.



Interaction with **DBI** is performed only via an external utility, path to which is specified in the **PathToDrwebQp** parameter ([\[Quarantine\] section](#)).

By default, **drweb-qp** is used. The utility is supplied with **Dr.Web MailD** and resides in `%bin_dir` directory.

To use **drweb-qp**, **Perl** (not older than 5.0) and installed **Perl DBI** and **File::Temp** modules are required.

To use **DBI**, set *Yes* value for the **MoveToDBI** parameter and configure **DBISettings**, **DBIUsername** and **DBIPassword** correspondingly to enable access to **DBI** storage.

Moreover, you need to configure the following SQL commands to perform required actions:

- **SQLInsertCommand** - this command adds a mail message to the **DBI** storage.
- **SQLRemoveCommand** - this command removes a mail message from the **DBI** storage. It is used when limitation on time to a message in **Quarantine** is set.
- **SQLSelectCommand** - this command provides access to the message stored in the **DBI** storage. It is used when a message is requested (for example, via the [control message](#)) from **Quarantine**.



Possible problems:

If you encounter an error of the following type:

```
maild ERROR Error in system call for [/opt/drweb/drweb-qp --Level debug --
SyslogFacility Daemon --BaseDir /var/drweb/ --ProcessMail 1 --MoveToDBI 0
--StoredTime 86400 --SQLInsertCommand "" --MDCClient "def" >/dev/null 2>&1
&]
```

try to increase the maximum available amount of memory for `drweb-maild` process (for example, using `ulimit -m` command).

Using Control Email Messages

Access to **Quarantine** can be gained via special *control email messages*. These messages, in the `Subject` field, contain commands to be executed by `drweb-maild`. Messages must be sent to the address specified in the `FilterMail` parameter from the [Notifier] [section](#) of the **Dr.Web MailD** configuration file or in local [rules](#) which are true for this message. ACL setup for control messages is performed by setting a value of the `OnlyTrustedControlMails` parameter from the [Maild] [section](#) of the **Dr.Web MailD configuration file**.

Receiving messages from **Quarantine** is enabled by setting `Yes` as a value of the `AccessByEmail` parameter from the [Quarantine] [section](#) of the **Dr.Web MailD** configuration file.

To receive a certain message from **Quarantine**, specify the following string in the `Subject` field of the control message:

```
q:relative_path_to_file
```

where `relative_path_to_file` is a relative path to the file in **Quarantine** (for example, `/def/drweb/F/00014F8F.DW_SHOT_PRODUCT.QqFpdH`).

The message is requested from **Quarantine** only if its sender or one of its recipients is the same as the sender of a *control message*.

The control message can be automatically generated by the user MUA when the user clicks the corresponding link in report, sent by **Dr.Web Notifier** upon transfer of message to **Quarantine**.

Migrating to New Quarantine Version

Dr.Web MailD version 6.0 and later features **Quarantine** of a new format: message body is saved to the file system, but message envelope and additional information are now saved to the internal database.

To upgrade **Quarantine** to a new format (for **Dr.Web MailD** version 5.0 and later) use the special `quarantine_migration.pl` script located in the `%bin_dir/maild/scripts/` directory. After startup, the script determines default settings and prompts to start migration to the new **Quarantine** version. Migration is performed fully automatically. After migration, the script outputs information on the operation results: start and end time and number of processed, skipped messages, and those that caused errors.

Interactive Management

During **Dr.Web for UNIX mail servers** operation, interactive management of some of its modules can be performed.



To enable interactive management

1. Set `Yes` as a value of the `Control` parameter in the [Maild] [section](#) of the **Dr.Web MailD** configuration file;
2. Connect to the address set in the `ControlAddress` parameter of the same section of the **Dr.Web MailD** configuration file, and start entering required commands (use any text client, for example, **telnet**).

Interaction is performed row-wise: the user inputs a string and `drweb-maild` responds to it. So, multiline commands are not supported and complex [Rules](#) must be entered line by line.

An empty line indicates the end of the output from `drweb-maild`.

Several interactive connections can be established simultaneously. Both IPv4 and IPv6 protocols are supported. **Dr.Web MailD** always opens listening socket at `/<value_of_BaseDir_parameter>/ipc/.ctl`, regardless of the `Control` parameter value.

Interactive management control socket allows you to:

- [Manage the general status](#) of the solution;
- [Manage users](#), join them in user groups, create aliases and adjust parameters of messages processing;
- [Manage Quarantine](#);
- [Get gathered statistics](#);
- [Check Notifier](#) operation.

General Management Commands

General commands and their descriptions are presented in the following table:

Command	Description
<code>help</code> [<section_name> <command_name> all]	Outputs complete list of commands from all the sections. After this command, you can specify the section name to receive information on all of the commands from it. Also you can specify the name of the specific command to get information only on it. You can view the list of all commands with the <code>help all</code> command.
<code>option</code> [regex]	Outputs the list of the parameters and their values used by both <code>drweb-maild</code> and plug-ins (which received their settings from Dr.Web MailD). The output parameter names match the specified regular expression. If a regular expression is not specified, all parameters are output.
<code>db-state</code>	Outputs the current state of the Dr.Web MailD internal database in the following format: Number: NC/NM Size: SC/SM where NC and NM are current and maximum amount of messages in the database, and SC and SM are current and maximum size (in bytes) of the database. If NM or SM are equal to 0, the maximum amount of messages and maximum database size are not limited (you can set the restrictions in the settings).
<code>queue-state</code>	Outputs the current status of messages in the internal queue. Total amount of messages and information about each message are displayed. If the total amount is large, that may indicate a lack of threads in the second pool of <code>drweb-maild</code> (regulated by the <code>OutPoolOptions</code> parameter)
<code>send-stat</code>	Enforces transfer/export of the statistics (similar to the action performed on timeout)



Command	Description
	<p>specified in the SendPeriod parameter from the [Stat] section of the Dr.Web MailD configuration file).</p> <p>This command can be used if the value of the Send parameter from the [Stat] section of the Dr.Web MailD configuration file is set to Yes. Gathered statistics is transferred to Dr.Web Agent.</p>
send-report [period]	<p>Enforces sending of reports on plug-in operations (similar to the action performed on timeout set in the SendPeriod parameter from the [Reports] section of the Dr.Web MailD configuration file).</p> <p>This command can only be used when the value of the Send parameter from the [Stat] section of the Dr.Web MailD configuration file is set to Yes.</p> <p>Period defines a time period for which the report is generated (value is in the {time} format). If this value is not specified, the report contains statistics for a 24-hour period.</p>
backup	Enforces back-up of the internal database
quarantine-pulse	Enforces start of the drweb-gp utility for Quarantine processing (similar to the action performed on timeout set in the PulseTime parameter from the [Quarantine] section of the Dr.Web MailD configuration file)
dump-cache-stat	All cached statistics is moved from cache to the internal database
get [(id1 - id1-[id2]) [(plugin_name -)]]	<p>Outputs information on messages stored in the internal database.</p> <p>where:</p> <ul style="list-style-type: none">• id – number of the requested message,• id1-id2 – outputs information on messages with numbers from the range defined by these values,• id1- – outputs information on all messages with numbers beginning with id1 (numbers must be specified in hexadecimal notation),• plugin_name – name of the plug-in, which moved the message to the database. <p>"-" symbol means that the parameter is not specified. When no parameter is specified, information on all messages from the database is output.</p> <p>Example:</p> <p>get - drweb – outputs information on messages moved to the database by Drweb plug-in.</p> <p>get - – outputs information on all messages stored in the database.</p>
send [(id1 - id1-[id2]) [(plugin_name -)]] [force]]	<p>Sends specified messages to the recipients from the envelope.</p> <p>Only messages not yet dispatched to the recipients (with send=no in get command output) can be sent. Description of the parameters is similar to that of the get command except for the new parameter - force, which initiates dispatch of messages with send=yes.</p>
export [(id1 - id1-[id2]) [(plugin_name -)]] [(dir_name -)]] [env]	<p>Exports specified messages from the database to external files.</p> <p>Description of the parameters is similar to that of the get command except for the following two additional parameters:</p> <ul style="list-style-type: none">• dir_name – path to the directory where files are stored. If this path is not specified, the value of the BaseDir parameter from the [General] section of the Dr.Web MailD configuration file is used;• env – if specified, the envelope is also exported to file in the following format:<ul style="list-style-type: none">○ first line – the sender address;○ second line – receiver addresses, separated by commas. <p>Name of the message file is created from the number of the message and .eml extension. Name of the envelope file is created from the number of the message and</p>



Command	Description
	.envelope extension. Example: <code>export 00002D94 vaderetro /t env</code> Success export body to /t/00002D94.eml and envelope to /t/00002D94.envelope
remove [(id1 - id1-[id2]) [(plugin_name -)]]	Removes specified messages from the database. Description of the parameters is similar to that of the get command. Example: <code>remove 00002D93</code> Success remove record 00002D93
send_and_remove [(id1 - id1-[id2]) [(plugin_name -)]] [force_send] [ignore_send_error]	Sends and removes specified messages from the database. Value of the <code>force_send</code> parameter is similar to the value of the <code>force</code> parameter of the send command. If the message was successfully sent with the send_and_remove command, or dispatch of this message is not required (that is, the message was sent before), the message is deleted. If the <code>ignore_send_error</code> parameter is specified, the message is deleted regardless whether the dispatch was successful or not.
version	Outputs the current product version.
stop	Stops the product operation.
reload	Sends <code>SIGHUP</code> signal to <code>drweb-maild</code> process.

User, Group, and Alias Management

User, Group, Alias concepts

User in **Dr.Web for UNIX mail servers** is an owner of one or more mailboxes whose mail correspondence must be processed according to special settings. If a user has more than one mailbox, their addresses are called *aliases* (at that, one of the addresses is treated as primary).

You can specify individual [Rules](#) of message processing for a user (similar to the way general Rules are specified in the `[Rules]` [section](#) of the configuration file). User address (or any alias) is considered to be the user identifier. All user addresses are treated as a unit, so the same settings are applied to them and their statistics is aggregated.

Users can be joined in *user groups* which can also have their own rules of messages processing. Each user can be included in any number of groups. You can set the following two flags for a user (or a group):

- **Activity flag (A)** – determine if the user (group) is active. If yes, special settings are used when processing. Otherwise, the special settings are ignored.
- **Statistics flag (S)** – enables or disables statistics gathering for the user (group). To enable gathering of statistics for an individual user, set level of [general statistics gathering](#) to `high`.

Parameter search algorithm

To determine a parameter value for a processed message, the following algorithm is used:

- Parameter value is searched in [Rules](#) stored in the [internal database](#) and related to the message **recipient** (the recipient is specified by the sender in `RCPT TO`).
- Parameter value is searched in Rules stored in the internal database and related to all groups of the recipient. Viewing of the rules is performed in reverse order: from rules of the last group in the list to the rules of the first one until the required value is found.
- Parameter value is searched in the `[Rules]` [section](#) of the main [configuration file](#).



Note the order of viewing Rules:

- All Rules in the currently viewed group are checked in the order they are specified.
- For each Rule, the `CONDITION` part is checked. If it is true, the required parameter value is searched in the `SETTINGS` part of the rule.
- If the `CONDITION` is false, the parameter value is searched in the next Rule.
- If the `CONDITION` is true and is followed by the `cont` directive, the parameter value is searched in the next Rule. If the `stop` directive is specified, viewing of Rules stops regardless whether the required value is found or not.

According to the results, the parameter value is determined in the following way:

- If the searched parameter is found in one matching Rule, its value from the `SETTINGS` part is used (note that when several Rules match the same parameter, the resulting value depends on the parameter semantics. For details, see [Message processing rules](#)).
- If no Rule is specified, no Rule is matching, or no matching Rule contains the required parameter value, it is retrieved from the corresponding section of the configuration file.
- If the parameter is not specified in the configuration file, the default value is used.

Thus, order of specifying user groups is important because it determines options to be applied to the given address.

If a message is sent to more than one recipient and for different recipients different values of the same parameter are specified, one of the following is possible:

1. The message is cloned and for each copy the corresponding parameter value is applied.
2. If the parameter does not allow cloning, the value is taken from user settings, or global settings specified in the configuration file, or the settings specified by default.



When searching the value, all the rules (for a certain user or user group) are viewed as a single list (user rules are at the top of the list, group rules – at the end of it). Thus, when viewing the lists for different message recipients, a user setting for a recipient can match a group setting for another recipient. If so, the algorithm described above is used.

You can use the control socket interface as well as the web interface for work with users, user groups, and aliases.

Interactive control socket commands for viewing users and user groups

- **email-info**

Show all information about users. As well as individual Rules of message processing, the internal database stores additional information on each user. The information can be output in the following format:

```
[client-id/]email A=0|1 S=0|1
name: username
aliases: alias1 alias2 ...
groups: group1 group2 ...
rules:
1: SETTINGS1
2: SETTINGS2
...
custom:
tag1: info1..
tag2: info2..
...
```

where:

- `client-id` – an empty string;



- **A** - flag indicating whether the user is active. If the user is inactive, all related Rules are ignored;
 - **S** - flag indicating whether gathering of individual statistics is enabled (at that, level of [general statistics gathering](#) must be set to `high`);
 - **name:** `name1` - user name (used mainly in the web interface);
 - **aliases:**, **groups:**, **rules:**, **custom:** - lists of aliases, user groups, rules and other user settings.
- **groups-info**

Show all information about user groups. Each group has the same set of options as that for a user: always empty `client-id` string, user group name, activity status, statistics flags, list of users and additional service information. Output format is as follows:

```
[client-id1/]group A=0|1 S=0|1
emails:
email1
email2
...
custom:
tag1: info1..
tag2: info2..
...
```

Interactive control socket commands for managing users and user groups

Managing of users, user groups, and aliases is performed with special commands in which the following concepts are used:

- **email** - user mail address (according to [RFC 5322](#)). It can be enclosed in angle brackets (`<>`) or single quotation marks (`'`). Address length cannot exceed 1024 bytes.
- **client-email** - pair of `[client-id/]email` values, where `client-id` for **Dr.Web MailD** is always an empty string.
- **emails-list** - List of **client-email** pairs. Items are separated by whitespace.
- **group** - Group name enclosed in single quotation marks (`'`). If a group name does not contain spaces, the quotation marks can be omitted. If a group name is enclosed in single quotation marks, `'` symbol within the group name must be doubled, (for example: `'It's a group name' -> 'It''s a group name'`). Length of a group name cannot exceed 1024 bytes.
- **client-group** - pair `[client-id/]group` values, where `client-id` for **Dr.Web MailD** is always an empty string.
- **ext-client-group** = `[client-id/]group | client-id/` - similar to **client-group**, where `client-id` for **Dr.Web MailD** is always an empty string.
- **group-list** - List of **client-group**. Items are separated by whitespace.
- **ext-group-list** - List of **ext-client-group**. Items are separated by whitespace.
- **RULE** - [Message processing rule](#). If the value contains a comma and this symbol is not enclosed in quotation marks, the `'\'` symbol must be specified before the comma (only if the value is not a comma-separated list; otherwise specify three back slashes `"\\\"`).

Examples:

```
true cont headersfilter/RejectCondition = FileName = "\"\.e\\\",e\"\",
FileName = "\"\.com\"\", headersfilter/RejectPartCondition = FileName =
\"\.e\\\",e\"\", FileName = "\"\.com\"\"
true cont vaderetro/action = discard\", quarantine
```

- **tag** - a string that contains symbols from the `[a-zA-Z0-9_-]` set. The string is used for search of information related to the user or user group. For the web interface, the value is set to



web.

- **info** - a string that contains information on the user or user group. Cannot contain line breaks and null symbols.
- **settings** - set of settings specified for the user (user group). Specified as a list of `parameter=value` pairs separated by whitespace.

The following parameters are allowed in the current version:

- **A (active)** - can be set to 0 (not active) or 1 (active). If the user or user group is not active, the rules related to them are ignored when processing. By default (if the flag is not specified), users and user groups are active.
- **S (stat)** - configures gathering of statistics for the user or user group. The value can be set to 0 (disabled gathering) or to 1 (enabled gathering). Setting flag to 0 stops statistics gathering. If statistics is already saved to the internal database, the data is accessible and is not deleted. By default, statistics gathering is enabled.
- **N (name)** - extended user name (this parameter is ignored for groups). Can be enclosed in single quotation marks (similar to **group**). If not specified, the user name is empty. The name length cannot exceed 1000 bytes.

Examples:

```
S=1 A=0 N='Some user'  
S=0
```

Please note that in order to support the sequence of groups for a certain `client-email`, management is performed for a group set for a `client-email`, but not for a set of `client-emails` for a group.

Commands for User Management

When operating via the control socket, a term *user* refers to every email address entered in the system. Addresses can be managed with the following commands:

- **email-set** `client-email [settings]` - create or update an email address, set in `client-email`. If the address does not exist, it will be created. For parameters that are not specified in `settings`, their default values are set. If an alias is set for the address, you can specify this alias as a `client-email` value upon an update.
- **email-remove** `client-email` - remove an email address, set in `client-email`. User is also deleted from all user groups. If the address does not exist or an alias is specified, an error is output.
- **email-rename** `client-email email` - change the main user address, set in the first parameter, to the address, set in the second parameter. If the address specified in the first parameter does not exist, or is alias, or an address with the new name already exists, an error is output and no action is performed.
- **email-set-groups** `client-email [list-of-groups]` - set the list of groups which contain `client-email` address. The group order is important (group settings at the end of the list prevail).

If `list-of-groups` is empty, the whole list of groups for `client-email` address is discarded. In `list-of-groups`, groups are space delimited. If `client-email` or another group in the list does not exist, an error is output and no action is performed. If a group is found more than once in the list, an error is output. If `client-email` is alias, the original recipient is updated. If `client-id` is specified for the `list-of-groups` list, it must match `client-id` from the `client-email` address. Otherwise, an error is output. If `client-id` is not specified in alias from `list-of-groups`, it is considered equal to the `client-id` from the `client-email`.

- **email-get-groups** `emails-list` - receive the group list with all the addresses from



`emails-list` list. If an address from the list is missing, an error is output but the execution continues. If `client-email` is alias, information on the original recipient is output.

Output format:

```
client-id/email1: group1 group2 group3 ...
client-id/email2: group21 group22 group23 ...
```

where `groupN` may be enclosed in single quotation marks, if the group name contains white spaces.

- **email-get-rules** `emails-list` - receive Rules for all the addresses from the `emails-list` list. If an address is not in the list, an error is output but the execution continues. If an alias is transmitted, settings for the original recipient are output. For every nonexistent address, an error is output.

Output format:

```
[client-id1/]email1
1: rule1
2: rule2
...
[client-id2/]email2
1: rule21
2: rule22
...
```

- **email-insert-rule** `client-email index RULE` - insert a new Rule before the Rule with `index` ordinal number for the email address specified in `client-email`. If an email address does not exist, an error is output. Numeration (`index`) starts with 1. If the `index` value exceeds the maximum number of rules for the specified email, the new `RULE` is added to the end of the rule list. At that, next ordinal number is assigned to the new rule as `index`.

Example: When adding a rule with the `index = 10` to the list consisting of only 2 rules, the new one is added to the end of the list and `index=3` is assigned to it. If the `index` is less or equal to 0, an error is output. If the `RULE` is empty (that is, the rule is not specified), an error is output.

After successful modification, the rules for the current group are output in the **email-get-rules** format.

- **email-remove-rule** `client-email index` - remove the rule with `index` ordinal number specified for the email address in `client-email`. Numeration (`index`) starts with 1. If `client-email` does not exist, an error is output. If the `index` value exceeds the maximum number of rules for the specified email, or `index` is less or equal 0, an error is output. If the specified value is alias, settings for the original address are updated.

After successful modification, the rules are output in the **email-get-rules** output format.

- **email-get-custom** `-tag emails-list` - receive information with the `tag`, associated with each user in the `emails-list`. If an address does not exist, an error is output but the execution continues. If information with the `tag` is not found, an empty string is output. Information for each address is separated by a line feed. If the "-" symbol is specified instead of `tag`, information on all tags is output.

Output format:

```
[client-id1/]email1
tag: info..
[client-id2/]email2
tag2: info2..
```

- **email-set-custom** `tag client-email [info]` - set the `info` text, connected with the `tag` for the `client-email` user. If the user is not found, an error is output. If `info` is not specified, the tag with all the related information is deleted.
- **email-info** `emails-list` - receive complete information on all addresses in the `emails-`



list. If an address does not exist in the list, an error is output, but the execution continues. Output Rules for an address are compiled for all groups and address settings. If an alias is specified, the information on groups and settings is taken from the original address. Rule settings are output in the following order: at first - user settings and after them - group settings in the order reversed to the group sequence. When compiling the rules, activity settings for the group and users is considered.

Output format:

```
[client-id1/]email1 A=active1 S=stat1
name: name1
aliases: alias1 alias2 ..
groups: group1 group2
rules:
1: rule11
2: rule12
...
custom:
tag1: info1..
tag2: info2..
...
[client-id2/]email2 A=active2 S=stat2
name: name2
aliases: alias12 alias22 .. | alias for email2
groups: group3
rules:
1: rule21
2: rule22
...
custom:
tag21: info21..
tag22: info22..
...
```

groupN may be enclosed in single quotation marks, if the group name contains spaces.

Output format for alias:

```
[client-id1/]email1
aliases: alias for email
```

- **email-search** [range:START/NUMBER] [email:part-of-email] [name:'part-of-name'] [ignore:alias|nonalias] - search by address or part of the address. It outputs the addresses starting with START (numeration begins with 0) in quantity of NUMBER elements. If the START and NUMBER values are not specified, all found addresses are output. If START or NUMBER are negative, an error is output. If values of START or NUMBER exceed the number of found addresses, their values are treated as not limited (thus, for "unlimited" START addresses are output from the first one in the list, for "unlimited" NUMBER all addresses in the list are output).
 - part-of-email - substring in the email address or alias used for the search. If part-of-email is not specified, all known addresses or aliases are output. Output format is similar to the one of **email-info** output. User unique identifier in part-of-email must be fully specified.
 - part-of-name - substring in the user name (if the name contains a single quotation mark ', another single quotation mark ' must be specified before it; if substring contains no spaces, enclosing quotation marks can be omitted). Only users whose names contain the specified substring are output.
 - ignore - defines type of records to be ignored - alias (search is performed among usual addresses), nonalias - usual addresses (search is performed among aliases).

If email and name are specified simultaneously, only those users are output that satisfy both



restrictions. As the user name is not saved for aliases, it is meaningless to use the substring for alias and user name simultaneously when searching.

- **email-count** [range:START/NUMBER] [email:part-of-email] [name:'part-of-name'] [ignore:alias|nonalias] - processing is performed similar to **email-search**, but only the number of found addresses is output.

Commands for Alias Management

You can manage aliases with the following commands:

- **aliases-get** emails-list - output list of aliases for all addresses from the emails-list list. If emails-list contains nonexisting addresses or other aliases, an error is output but the execution continues. If the same address is found more than once, an the error is output.

Output format:

```
[client-id1/]email1: alias1 alias2 alias3 ...  
[client-id2/]email2: alias21 alias22 alias23 ...
```

- **aliases-set** client-email [emails-list] - set the list of aliases for email address, specified in client-email. If client-email does not exist or is alias, an error is output. If emails-list is not specified, all aliases, related to client-email, are deleted. If the list contains at least one registered address or alias for another address, an error is output and the execution stops. If client-id is specified for the address in emails-list, it must match client-id from client-email address. Otherwise, an error is output. If client-id is not specified in emails-list in alias, the id is treated as equal to client-id which is set in client-email.

Commands for Group Management

You can manage user groups with the following commands:

- **groups-set** client-group [settings] - create or update the group with the group name, set in client-group. If the group does not exist, it will be created. For parameters that are not specified in the settings, their default values are used.
- **groups-remove** client-group - remove the group with the group name, specified in client-group. If the specified group does not exist, an error is output. For every user who is a member of the removed group, it is deleted from the user group list.
- **groups-rename** client-group group - rename the group, specified in the first parameter, to the name specified in the second parameter. If the specified group does not exist or the group with the new name already exists, an error is output and no action is taken.
- **groups-get-rules** [group-list] - receive the rules or settings for all groups from the group-list list. If a group from group-list does not exist, an error is output but the execution continues.

Output format:

```
[client-id1/]group1  
1: rule1  
2: rule2  
...  
[client-id2/]group2  
1: rule21  
2: rule22  
...
```

- **groups-insert-rule** client-group index RULE - insert a new rule before the rule with the index ordinal number for a group with the group name, specified in client-group. If the group with the specified name does not exist, an error is output. Numeration (index) starts with



1. If the `index` value exceeds the maximum number of rules for the specified group, the new `RULE` is added at the end of the rule list. At that, next ordinal number is assigned to the new rule as `index`.

Example:

When adding a rule with the `index = 10` to the list consisting of only 2 rules, the new one is added to the end of the list and `index=3` is assigned to it.

If the `index` is less or equal to 0, an error is output. If the `RULE` is empty (that is, the rule is not specified), an error is output.

After successful modification, the rules for the current group are output in the `groups-get-rules` format.

- **groups-remove-rule** `client-group index` - remove the rule with `index` ordinal number for the `group`, specified in `client-group`. Numeration (`index`) starts with 1. If a `group` does not exist, an error is output. If the `index` value exceeds the maximum number of rules for the specified group or `index` is less or equal to 0, an error is output. After successful modification, the rules for the current group are output in the `groups-get-rules` format.
- **groups-info** [`ext-group-list`] - output all users from `ext-group-list` list, information about their activity, and other information. If a group from `ext-group-list` does not exist, an error is output but the execution continues. If `ext-group-list` is not specified, information on all existing groups is output. Aliases in the address lists are not output.

Output format:

```
[client-id1/]group1 A=active1 S=stat1
emails:
email1
email2
...
custom:
tag1: info1..
tag2: info2..
...
[client-id2/]group2 A=active2 S=stat2
emails:
email21
email22
...
custom:
tag21: info21..
tag22: info22..
...
```

- **groups-count** [`ext-group-list`] - this command is similar to `groups-info` command, but outputs only the number of found groups.
- **groups-get-custom** `-|tag group-list` - receive information with the `tag`, associated with each user in the `group-list` list. If a group does not exist, an error is output but the execution continues. If information with the `tag` is not found, an empty string is output. Information for each group is separated with a new line. If the "-" symbol is specified instead of `tag`, information on all tags is output.

Output format:

```
[client-id1/]group1
tag: info..
[client-id2/]group2
tag2: info2..
```

- **groups-set-custom** `tag client-group [info]` - setting of the `info` text, connected with the `tag` for `client-group` group. If the group is not found, an error is output. If `info` is



not specified, the tag with all the information, related to it, is deleted.

Working with Quarantine

You can manage **Quarantine** via the control socket using special commands. In these commands, the following common notions are used:

- **id** - relative path from the directory, specified in the **Path** parameter in the [Quarantine] [section](#), to the file with the text body.
For example, if the **Path** parameter value in the [Quarantine] section is `/var/drweb/infected` (default value), then `<id>/drweb/E/00020EBE.maild.xeAX4u` path refers to the message which body is located in the file `/var/drweb/infected/<id>/drweb/E/00020EBE.maild.xeAX4u`, where:
 - `<id>` - the "def" string;
 - `drweb` - name of the [plug-in](#) that blocked the message (**Drweb** in this case). If the message is blocked by **Mails core**, the value is set to `maild`. If the message is moved to archive, the value is set to `backup`.
- **id-like** - the same as **id**, but in identifiers of this type, special symbols can be used: `"%"` - zero or more random symbols, `"_"` - one random symbol.

Example:

`def/%00014F7F%` - all messages with `00014F7F` number, saved to **Quarantine**;

`def/drweb/%` - all messages saved by **Drweb plug-in**.

Message body is saved to the database in the decoded form (in UTF8 encoding), and all control characters (ASCII 0..21 and 127), except for tabs, are replaced with spaces.

Execution results are output with an empty string at the end.

Commands for Quarantine Management

You can manage **Quarantine** with the following commands:

- **quarantine-search** [range:START/NUMBER] [sort:SORT_TYPE] [sender:EMAIL_SUBSTR] [rcpt:EMAIL_SUBSTR]* [period:DATE1[/DATE2]] [size:SIZE] [subject:'SUBJECT_SUBSTR'] [id:id-like] [order:asc|descent] - searches messages in **Quarantine** by the specified criteria. Messages are output starting with `START` (numeration begins with 0), and in the quantity of `NUMBER` elements. If `START` and `NUMBER` are not specified, all found messages satisfying other criteria are output. If the `NUMBER` value is set to 0, all elements are output.

The following parameters are used:

- `SORT_TYPE` - type of the sorting. Possible values:
 - `date` (by default) - sort by date of moving messages to **Quarantine**;
 - `size` - sort by message size;
 - `sender` - sort by sender address;
 - `subject` - sort by message subject.
- `EMAIL_SUBSTR` - substring for search in the `rcpt` or `sender` fields.
- `period` - period for which messages are output. If not specified, messages are output for the whole period.
- `DATE1` - output messages moved to **Quarantine** after the specified time (inclusive).
- `DATE2` - upper time line of moving the message to **Quarantine** (inclusively). The `DATE` format corresponds to the ISO format - `YYYYMMDDTHHMMSS`, where `T` - separator between



time and date. The time is set and output as local time on a host where **Dr.Web MailD** is operating.

- **SIZE** - maximum size (in bytes). Returns only messages with the size exceeding the specified value. When the value is set to 0, the size is unlimited.
- **SUBJECT_SUBSTR** - substring enclosed in quotation marks in the original subject of the message (that is, before the message was modified by the product components). If the substring does not contain spaces, the enclosing quotation marks can be omitted. If quotation marks are present in the name, the ' character must be doubled.
- **order** - order in which results are returned (**ascent** - ascending, **descent** - descending). Default value: **descent**.

If there is a mistake in a parameter, the **quarantine-search** command cannot be executed. If several recipient templates are specified, only messages that contain all templates are output (similar to the effect of **AND** logical operator). For all parameters, except **rcpt**, the value specified last in the command line is used. For **rcpt**, the number of recipients increases with every new input value.

Output format:

```
N. id SENDER RCTPS
SIZE DATE SUBJECT
BLOCK_OBJECT1
BLOCK_OBJECT2
...
```

where:

- **N** - ordinal number of the found message;
- **SENDER** - sender of the message from the envelope;
- **RCPTS** - recipients of the message from the envelope;
- **SUBJECT** - subject of the message (output in UTF8 encryption);
- **SIZE** - size of the message, in bytes;
- **DATE** - date of moving the message to **Quarantine**;
- **BLOCK_OBJECTN** - object blocking the message.

Examples:

```
# quarantine-search
```

Returns the list of all messages in **Quarantine**, starting from the newest.

```
# quarantine-search range:45/15 id:def/drweb/%
```

Returns 15 newest messages in **Quarantine**, missing first 45 messages for the **Drweb** plug-in.

```
# quarantine-search rcpt:john@smith.com
```

Returns all messages with **john@smith.com** recipient in **Quarantine**, starting from the newest.



```
# quarantine-search sort:size sender:
period:20090101T100001/20090102T100000 size:5242880 id:def/vaderetro/%
```

Outputs the messages in descending order, received for **Vaderetro plug-in** on January 1, 2009 from 10 a.m. to 10 a.m. of the next day, and size of which exceeds 5 MB.

Output example:

```
# quarantine-search
0.      def/drweb/9/00021569.maild.BMED3y      <ai@drweb.com>
<alias_ai81@drweb.com>
829 20091117T102126 [EICAR] test2
EICAR Test File (NOT a Virus!)
1.      def/backup/9/00021569.maild.3PLb8e      <ai@drweb.com>
<alias_ai81@drweb.com>
828 20091117T100213 [EICAR] test
```

- **quarantine-count** [range:START/NUMBER] [sort:SORT_TYPE] [sender:EMAIL_SUBSTR] [rcpt:EMAIL_SUBSTR]* [period:DATE1[/DATE2]] [size:SIZE] [subject:'SUBJECT_SUBSTR'] [id:id-like] [order:ascent|descent] - similar to the **quarantine-search** command, but, instead of the messages, total number of found messages is output.

Output examples:

```
# quarantine-count
234
```

- **quarantine-remove** id-like [part-of-email1, part-of-email2, ..] - removes the specified recipients (searched as a substring) part-of-email1, part-of-email2, ... from the envelopes of the messages, identifiers of which match id-like (all specified recipients must be present in the envelope). If a message has no recipients left or the list for their removal is not specified, the whole message is removed from **Quarantine**.

Examples:

```
# quarantine-remove %/backup/% drweb.com>
```

All messages sent to addresses ending in `drweb.com` are deleted from **Quarantine** and backup.

```
# quarantine-remove % <foo@dwreb.com> <foo2@dwreb.com>
```

All messages sent to both `foo@dwreb.com` and `foo2@dwreb.com` addresses are deleted from **Quarantine** and backup.

- **quarantine-limits** - outputs the current restrictions set for **Quarantine**.

Output format:

```
client-id: NUMBER/MAX-NUMBER SIZE/MAX-SIZE
...
total: NUMBER/MAX-NUMBER SIZE/MAX-SIZE
```

where:

- NUMBER/MAX-NUMBER - current/maximum number of messages. If the maximum value is not set, 0 is output.
- SIZE/MAX-SIZE - current/maximum size of messages in **Quarantine** (in bytes). If the maximum value is not set, 0 is output.
- client-id1 - "def" string.
- total - information on the whole database.



- **quarantine-send** id-like [email1 email2 ...] - sends messages from **Quarantine** to the specified recipients (email1 email2 ...). If no recipient is specified, the messages are sent to their original recipients from the envelope. Message sending result is output in the following format:

```
RES in sending (to RCPTS_LIST): id
```

where:

- RCPTS_LIST - actual list of message recipients.
- RES - OK or ERROR, depending on the sending result.
- id - path to the file with the message body.

Output format:

```
OK in sending (to <ai@drweb.com> <as@sd>): def/  
backup/6/00004DD6.maild.VQ80Ro  
OK in sending (to <ai@drweb.com> <as@sd>): def/  
backup/6/00004DC6.maild.PWfqe3
```

- **quarantine-add** id from rcpt1 rcpt2... - adds the specified file to **Quarantine**, where from - message sender, rcptN - recipients. Addresses can be enclosed in angular brackets <>. If a file with the specified id does not exist, an error is output.

Receiving Statistics

Statistics on **Dr.Web for UNIX mail servers** operation for users and user groups can be received via the command interface of the control socket. Statistics is received with the use of special commands where the following common notions are used:

- **email** - user mail address (according to RFC5322). It can be enclosed in angular brackets (<>) or quotation marks (' '). Length of the address cannot exceed 1024 bytes.
- **client-email** - [client-id/]email, where client-id is always an empty string.
- **group** - name of the group enclosed in quotation marks (' '). If the substring contains no spaces, the quotation marks can be omitted. If a quotation mark is used, the character must be doubled. Length of the group name cannot exceed 1024 bytes.
- **client-group** = [client-id/]group - where client-id is always an empty string.

When working with statistics, consider that statistics on users and user groups is saved directly to the corresponding records of the internal database. When data is saved and the most recent record exists in the database for more than five minutes, a new record is created and further changes are saved to it.

As these commands work only with the internal database, the most recent record with statistics on users and user groups contains information for the period from its creation to the current moment.

Commands for Working with Statistics

Commands for working with statistics have the following general parameters:

- **period** = period:DATE1[/DATE2] - outputs statistics for the selected period, including the time interval limits.

where:

- DATE1 - lower limit of the time interval. Output format and time format are described below.



- `DATE2` - upper limit of the time interval. If the parameter is not set, the current time is used. Time format is described below.

If the period is not set, then all available statistics is output.

- `ignore` = `ignore:total|block` - filtration of output statistics.
- `total` - do not output general statistics on checked messages.
- `block` - do not output statistics on blocked messages. If this parameter is not set, statistics of all types is output.
- `plugin` = `plugin:name` - outputs information on the specified plug-in, where `name` is the name of the plug-in. If the `plugin` parameter is not set, information on all plug-ins is output. If a non-existing plug-in is specified, an error is output and command is canceled. If `*` is specified, general statistics is output.

If several similar parameters are specified, statistics is output only for the last specified parameter.

The following commands are available:

```
stat-client - [period] [ignore] [plugin]
```

The command is used for receiving statistics.

Optional parameters can be set in random order.

```
stat-group client-group [period] [ignore] [plugin]
```

The command is used for receiving statistics on the group specified instead of `client-group`.

- If the group does not exist, an error is output and the command is aborted.

Optional parameters can be set in random order.

```
stat-email client-email [period] [ignore] [plugin]
```

The command is used for receiving statistics on the current user specified instead of `client-email`.

- If statistics on the specified address does not exist (for example, the address is incorrect), an empty string is output.
- If the address is alias, statistics on the main address is output.

Optional parameters can be set in random order.

```
stat-remove-client - [period] [ignore] [plugin]
```

Removal of statistics.

As a result, number of deleted records is displayed.

```
stat-remove-group client-group [period] [ignore] [plugin]
```

Removal of statistics on the group (`client-group`).

As a result, the number of deleted records is displayed.

```
stat-remove-email client-email [period] [ignore] [plugin]
```

Removal of statistics on the specific user (`client-email`).

As a result, the number of deleted records is displayed.

```
remove-old-stat [time]
```

Removal of all statistics on all groups and users, if it is older than the time, set in `time` (type `{time}`).

If the value is not set, all statistics **older than 24 hours** is removed.

**dump-cache-stat**

Import the internal cache of general statistics on **Clients** to the internal database.

This function is periodically called by the software itself. It is also called on receipt of a `SIGHUP` signal or shutdown.

Statistics Output

Statistics on operation of each **plug-in** is output in the following format, which consists of two parts:

1. General statistics on checked messages:

```
PLUGIN DATE [P][R][D][T][Q][RE][N][C][S][U][F][I][DI][DM][DSV][DC][DD]
[DSK][DAR][DE][DTA][DTD][DTJ][DTR][DTH][PS][RS][DS][TS][QS][RES][NS]
[CS][SS][US][FS][IS][WT]...
```

2. Statistics on blocked messages:

```
PLUGIN DATE FROM|- IP|- 'BLOCK1' TYPE1 'BLOCK2' TYPE2 ...
```

Where:

- **PLUGIN** - name of the plug-in on which statistics is output. If `*` is specified, general statistics displays (including statistics on messages which were not transmitted by any plug-in).
- **DATE** - time when the record was created. For general statistics on checked messages, it means the beginning of the time interval during which statistics is saved. End of the time interval is the beginning of a new record. If the following record does not exist, then 5 minutes are added to the beginning of the period. The format matches ISO format - `YYYYMMDDTHHMMSS`, where `T` - separator between time and date. Time is set and output as local time for the host with **Dr.Web for UNIX mail servers**.

The following values until `WT` inclusive are output in the **NAME=VAL** format, where **NAME** - name of the value (`P, PS...`), **VAL** - its numeric value. If some of these values are not specified, it is treated as equal to 0:

- `P/PS` - `<number>/<size in bytes>` of the messages for which the `pass` action was applied ;
- `R/RS` - `<number>/<size in bytes>` of the messages for which the `reject` action was applied;
- `D/DS` - `<number>/<size in bytes>` of the messages for which the `discard` action was applied;
- `T/TS` - `<number>/<size in bytes>` of the messages for which the `tempfail` action was applied;
- `Q/QS` - `<number>/<size in bytes>` of the messages for which the `quarantine` action was applied;
- `RE/RES` - `<number>/<size in bytes>` of the messages for which the `redirect` action was applied;
- `N/NS` - `<number>/<size in bytes>` of the messages for which the `notify` action was applied;
- `C/CS` - `<number>/<size in bytes>` of clean messages;
- `S/SS` - `<number>/<size in bytes>` of the messages marked as spam;
- `U/US` - `<number>/<size in bytes>` of the messages marked as unconditional spam;
- `F/FS` - `<number>/<size in bytes>` of the messages blocked by the filter;
- `I/IS` - `<number>/<size in bytes>` of the messages with viruses;
- `DI` - number of infected attachments;
- `DM` - number of attachments infected with modification of a known virus;
- `DSV` - number of attachments infected with an unknown virus;



- DC - number of cured attachments;
- DD - number of deleted attachments;
- DSK - number of attachments that were passed without anti-virus check for various reasons;
- DAR - number of attachments that were passed without anti-virus check because of restrictions on archives;
- DE - number of attachments which failed to be processed;
- DTA - number of attachments with adware;
- DTD - number of attachments with dialers;
- DTJ - number of attachments with jokes;
- DTR - number of attachments with riskware;
- DTH - number of attachments with hack tools;
- WT - time in milliseconds that plug-in spent on message processing.

For blocked messages, the list contains the following fields:

- BLOCK[12...] - name of the blocking object (for example, virus). It is enclosed in quotes in same way as it is done for groups (see above).
- TYPE[12...] - type of the blocked object. Name is used from **NAME** (see above). Available values: DI-DTH, F, S, U.
- FROM - sender of the message from the envelope.
- IP - IP address of the message sender.

Examples

1. Request for statistics on all processed messages (for all plug-ins):

```
> stat-client -
```

Command format corresponds to the [description](#); statistics output format is described [above](#).

2. Request for statistics on unblocked messages (for all plug-ins):

```
> stat-client - ignore:block
```

Example of statistics output:

```
* 20120307T145754 P=1 C=1 PS=194562 CS=194562
headersfilter 20120311T163250 R=4 F=4 RS=39848 FS=39848
headersfilter 20120311T164250 R=1 F=1 RS=539 FS=539
headersfilter 20120311T165400 P=1 F=1 PS=539 FS=539 WT=32
drweb 20120311T165400 R=1 Q=1 C=1 DE=1 RS=539 QS=539 CS=539 WT=13
headersfilter 20120311T165727 P=1 F=1 PS=539 FS=539 WT=32
drweb 20120311T165727 P=1 C=1 DE=1 PS=539 CS=539 WT=11
* 20120311T165953 P=1 U=1 F=1 DE=1 PS=539 US=539 FS=539 WT=51
vaderetro 20120311T165953 P=1 U=1 PS=539 US=539 WT=5
modifier 20120311T165953 P=1 C=1 PS=539 CS=539 WT=3
headersfilter 20120311T170453 P=1 F=1 PS=539 FS=539
drweb 20120311T170453 P=1 C=1 DE=1 PS=539 CS=539 WT=11
* 20120311T171208 P=1 U=1 F=1 PS=539 US=539 FS=539 WT=52
vaderetro 20120311T171208 P=1 U=1 PS=539 US=539 WT=5
modifier 20120311T171208 P=1 C=1 PS=539 CS=539 WT=3
headersfilter 20120311T171412 P=1 F=1 PS=539 FS=539 WT=33
drweb 20120311T171412 P=1 C=1 DE=1 PS=539 CS=539 WT=5
```



3. Request for statistics on blocked messages only (for all plug-ins):

```
> stat-client - ignore:total
drweb 20120406T194038 root@testlab-solaris.i.drweb.ru 10.3.0.91 'No such
file or directory' DE - DE 'No such file or directory' DE - DE Trojan.Grab
DI 'No such file or directory' DE - DE
drweb 20120406T194039 root@testlab-solaris.i.drweb.ru 10.3.0.91 'No such
file or directory' DE - DE 'No such file or directory' DE - DE Trojan.Grab
DI 'No such file or directory' DE - DE
drweb 20120406T194040 root@testlab-solaris.i.drweb.ru 10.3.0.91 'No such
file or directory' DE - DE 'No such file or directory' DE - DE Trojan.Grab
DI 'No such file or directory' DE - DE
```

4. Request for statistics only on messages which were blocked by [Drweb anti-virus plug-in](#):

```
> stat-client - ignore:total plugin:drweb
```

5. Request for statistics only on messages which were passed by [Drweb anti-virus plug-in](#):

```
> stat-client - ignore:block plugin:drweb
```

Checking Notification Generation

You can check how **Notifier** generates notifications from [templates](#). For that purpose, use the `notify` command in the interactive management interface.

Command format

The command has the following syntax:

```
notify type [mode] [-] [options]
```

where :

- `type` - notification type which is the same as the name of the used template file. It is allowed to use any type except for `report.msg`.
- `mode` - notification mode which can be one of the following:
 - `show` - generate a message from a template and output the notification text.

Format of the output command:

```
SIZE <FROM> <RCPT1> <RCPT2>...
BODY
```

where: `SIZE` - size of the `BODY` in bytes. If `BODY` is not finished with a new line character, the character is added (but not included in `SIZE`), `FROM` - message sender (from the envelope), `RCPT` - message recipients (from the envelope), `BODY` - message body. For description of the procedure how recipients and sender are determined, see below.

- `sync` - generate a message from the template and send via **Sender** in the synchronous mode (that is, **Sender** delivers the message before the result is returned). If the used **Sender** does not support the synchronous mode, the message is sent in the asynchronous mode (see below). According to the results, a string on successful or failed sending of the generated notification is output.
- `async` - generate a message from the template and send via **Sender** in the asynchronous mode (that is, **Sender** receives the message for processing, but can send it later). If the used **Sender** does not support the asynchronous mode, the message is sent in the synchronous mode (see above). **Sender** must support operation at least in one of the modes. According to the results, a string on successful or failed sending of the generated notification is output.



If the `mode` is not specified, it is set to `show`.

- `client-id` - **Client** identifier for notification processing. If the '-' is specified, the identifier is treated as empty. Thus, the `notify` command allows to operate with one **Client** only..
- `options` - list of macros required to be initialized for the template (assigned values are used when generating a message). The list format is a set of the following pairs:

```
NAME=VAL
```

where `NAME` - macro name (without the surrounding \$) and `VAL` assigned value enclosed in single quotation marks. The macro name and macro value are case insensitive. If the `VAL` value does not have empty spaces, the enclosing quotes can be omitted. If the `VAL` value is enclosed in quotes and the ' character is included in the value, the character must be escaped with the repeated ' character.

Between `NAME`, '=' and `VAL` white spaces must not be specified. If several `NAME=VAL` pairs with the same `NAME` are specified, the `NAME` macro is treated as a list when processing the template. If, according to the template, the macro cannot be a list but is specified several times in the command, either its first value is used or all of the values are combined using commas (which is incorrect in most cases).

After the command output, an empty line is always added.

Macros Initialization Method

In the list of macros for initialization, any of [these macros](#) can be used. For all macros, except for those described below, their default value is used.

Values of macros used in the reports are not set automatically (the macros with the `RP_*` name) . These macros must be explicitly initialized in a command. If a macro necessary for notification generation is not specified, the corresponding error is output to the **Notifier** log.

Apart from macros, you can also use the following indicators as initialization parameters:

- `_FROM` - sets a sender in the envelope (also used for searching parameter values in [Rules](#) that have `sender:` in the [conditional part](#)). For DSN templates, the sender is always empty. The method to determine a message sender is described below. If several values are specified, the first one is used.
- `_RCPTS` - sets message recipients (also used for searching parameters in Rules that have `rcpt:` in the conditional part). This parameter can have a list as a value. The method to determine message recipients (for an envelope) is described below.
- `_BLOCK` - sets names for malicious objects that blocked a virtual message for which notification is generated. This parameter can have a list as a value and is used in Rules that have `block:` in the conditional part.
- `_SIZE` - sets size of a virtual message for which notification is generated, in bytes. This parameter is used for searching parameters in Rules that have `size:` in the conditional part.
- `_SCORE` - sets a score of the message for which the notification is generated. This parameter is used for searching parameters in Rules that have `score:` in the conditional part.

Sender- and Recipient-Determining Mechanism

If special values of `_FROM` or `_RCPTS` are specified, the sender and recipients in the envelope are determined by these indicators only (except for DSN templates where the recipient is always empty). Otherwise, the following mechanism to determine sender and recipients is used:

For `welcome_user.msg` and `password_user.msg` templates:

- sender's address is determined according to the **AdminMail** parameter from Rules;
- recipients' addresses - from the `$RCPTS$` macro (thus, it must be initialized otherwise an error



occurs).

For `welcome_client.msg` and `password_client.msg` templates:

- sender's address is set according to the value specified first for the **AdminMail** parameter in the configuration, ignoring the matching Rules.
- recipients' addresses are determined according to the **AdminMail** parameter from the matching Rules.

For DSN (`dsn.msg` or `dsn_for_exchange.msg`) templates:

- sender's address is always empty
- if the `_FROM` parameter is specified, the values specified first fro this parameter is set as the sender's address, otherwise, recipients' addresses are determined by the `$RCPTS$` macro. If the `_FROM` parameter is not specified and the `$RCPTS$` macro is not initialized, an error occurs.

For other templates:

- sender's address is set aaccording to the **FilterMail** parameter value from the matching Rules.
- recipients' addresses - according to the `$RCPTS$` macro (thus, the macro must be initialized, otherwise an error occur).

Command Example Usage

Output text of a message generated as DSN for the `ai@drweb.com` and `test@drweb.com` recipients

```
notify dsn.msg show - _RCPTS=ai@drweb.com FULLHEADERS='From: <fake>'
_RCPTS='test@drweb.com'
```

Generate a message from the `admin_virus.msg` template and send the message in the synchronous mode

```
notify admin_virus.msg sync - RCPTS=test FULLHEADERS='From: <fake>'
```

Dr.Web MailD Utilities

The following utilities are included in **Dr.Web for UNIX mail servers** suite for ease in adjustment and management of **Dr.Web MailD**:

- `drweb-qp` – used for interaction between **Quarantine** and DBI. Cannot be launched manually
- `drweb-qcontrol` utility – used for management of **Quarantine** and quarantined files
- `drweb-lookup` utility – used for check of expressions used in **Lookup**
- `drweb-inject` utility – used for manual sending of saved email messages (including messages "lost" due to processing errors) via the **Sender** component or other installed mail transfer component.

drweb-qcontrol: Quarantine Management

The `drweb-qcontrol` utility allows managing **Quarantine** and perform a search through it. The utility interface can be used for search of messages saved to **DBI storage** or in files on the disk.

If **Quarantine** is saved in files on the disk, it is required to start the `drweb-maild` **module** before.

The utility is launched with the following command:

```
drweb-qcontrol [parameters] command [, command, ...] <identifiers>
```



1. Command line parameters

The following command line parameters are available:

Short case	Extended case	Arguments
-h	--help	
<u>Description:</u> Output short help information on the command line parameters to the console and exit		
	--version	
<u>Description:</u> Output information on the utility version and exit		
-v	--verbose	
<u>Description:</u> Output information on all utility actions to the console		
-l	--level	<verbosity level>
<u>Description:</u> Set the logging verbosity level. The following levels are available: Quiet, Error, Alert, Info, Debug		
-i	--ipc-level	<verbosity level>
<u>Description:</u> Set the verbosity level for IPC records (interaction with drweb-maild). The following levels are available: Quiet, Error, Alert, Info, Debug		
	--syslogfacility	<syslog flag>
<u>Description:</u> Set the subsystem type used by syslog service for message output (if this service is used for logging, see the next parameter description). The following types are available: Daemon, Mail, Local0, ..., Local7		
	--logfile	<file name>
<u>Description:</u> Set the name of the log file or syslog (if this service is to be used for logging)		
	--sendmail	<path to file>
<u>Description:</u> Set the path to the executable file of drweb-inject utility used for sending messages (by default, if the parameter is not specified, the %bin_dir%/drweb-inject path is used)		
-s	--socket	<path to file>
<u>Description:</u> Set the path to the Dr.Web MailD control socket (by default, if the parameter is not specified, the %var_dir%/ipc/.ctl path is used)		
	--agent	<path to file>
<u>Description:</u> Set the path to the Dr.Web Agent socket to receive configuration (by default, if the parameter is not specified, the %var_dir%/ipc/.agent path is used). If the switch is specified without a path, configuration from Dr.Web Agent is not requested.		
	--timeout	<time period>
<u>Description:</u> Set the maximum allowed time to wait for response from Dr.Web Agent when requesting configuration		

2. Commands

Commands are used to apply actions to messages selected from **Quarantine** by the specified criteria.



The list of messages is composed according to the unique identifiers, that are relative paths to the quarantined messages. To set the identifiers, you can use the following special template symbols:

- "%" – corresponds to the symbol sequence of zero or indefinite length;
- "_" – corresponds to one arbitrary symbol.

Note that every identifier must start with `def/`.

Example:

`def/%00014F7F%` – all messages moved to **Quarantine** and number of which contains 00014F7F sequence or is equal to it;

`def/drweb/%` – all messages moved to **Quarantine** by **Drweb plug-in**.

File identifiers are taken from the command line or from [search criteria](#) specified on the startup (see below). If the search criteria and file identifiers in the command line are specified simultaneously, they are joined. If neither an identifier in the command line nor search criterion is specified, the identifier is expected in the standard input.

The following commands are available:

- `--view` – view all messages with a certain identifier via the program specified in the `PAGER` environment variable. If no value is specified, the `cat` program is used.
- `--send` – send all messages with a certain identifier to the original recipients. For dispatch, the `drweb-inject` [utility](#) is used.
- `--redirect [list_of_rcpts]` – send all messages with a certain identifier to the addresses from the `list_of_rcpts` list. For dispatch, the `drweb-inject` [utility](#) is used.
- `--remove` – remove all messages with a certain identifier from **Quarantine**.
- `--stat` – output statistics on quarantined messages with a certain identifier.

Example:

```
drweb-qcontrol --stat def/%
1. def/backup/B/00014F8B.DW_SHOT_PRODUCT.U0dshM from: ai@1; to: ai@fff;
time: 2008-08-14 12:10:57
2. def/drweb/F/00014F8F.DW_SHOT_PRODUCT.QqFpdH from: ai@4; to: ai@fff;
time: 2008-08-14 13:00:50
3. def/backup/C/00014F8C.DW_SHOT_PRODUCT.A39xp7 from: ai@2; to: ai@fff;
time: 2008-08-14 13:00:50
4. def/backup/F/00014F8F.DW_SHOT_PRODUCT.tMi6W2 from: ai@4; to: ai@fff;
time: 2008-08-14 13:00:50
5. def/drweb/3/00014F93.DW_SHOT_PRODUCT.n9xPjU from: ai@3; to: ai@fff;
time: 2008-08-14 13:30:49
6. def/backup/3/00014F93.DW_SHOT_PRODUCT.ewYFVA from: ai@3; to: ai@fff;
time: 2008-08-14 13:30:49
7. def/backup/4/00014F94.DW_SHOT_PRODUCT.JQ3sLH from: ai@3; to: ai@fff;
time: 2008-08-14 13:30:49
```

Actions are applied in the order they are set, that is, you can specify several actions in one command.

Example:

```
drweb-qcontrol --send --remove def/backup/F/00014F7F.DW_SHOT_PRODUCT.yv4ro9
```

sends an email message with the `def/backup/F/00014F7F.DW_SHOT_PRODUCT.yv4ro9` identifier to the original recipients and then deletes the message from **Quarantine**.



If **Dr.Web MailD** is set up to store quarantined messages in **DBI** storage, the following additional SQL command must be specified in the command line:

- `--sql-remove-command` - allows to remove a message from **Quarantine** by the file identifier (the only parameter is the identifier).

Example:

```
drweb-qcontrol --sql-remove-command "DELETE FROM mail_export WHERE filename LIKE ?"
```

3. Message search

The `drweb-qcontrol` utility provides a simple interface for searching quarantined messages. The following search criteria are available:

- `--search-from {address}` - search by sender in a message envelope;
- `--search-to {address}` - search by recipient in a message envelope;
- `--search-headers {header_name[:value]}` - search in headers of the top level.
where `header_name` is the name of the target header (full compliance is required). If `value` is not specified, a header is searched only by its name. Otherwise, the value is searched within the headers as a substring. Searched header name and its value are case insensitive;
- `--search-inbody {string}` - performs a search of the specified substrings in the message body which is treated as a unit, without MIME decoding. Searched value is case insensitive.

Note that if `*`, `^`, `$` special symbols are used in values of the `--search-headers` or `--search-inbody` parameters, the symbols are to be escaped with a backslash `"\"`.

Example:

```
drweb-qcontrol --search-inbody \* --stat
```

Outputs statistics on email messages satisfying the specified criteria of search in the body.

Each of the criteria is checked independently, that is, they are combined in a disjunction (`OR`).

Example:

```
drweb-qcontrol --search-to addr1 --search-to addr2
```

searches for email messages with envelopes that contain `addr1` or `addr2` addresses.

Example:

```
drweb-qcontrol --search-from from@drweb.com  
--search-to to@drweb.com --search-headers  
"Subject: [SPAM]" --search-inbody "spam"
```

finds all quarantined email messages sent by `from@drweb.com`, or sent to `to@drweb.com`, or topic of which contains the `[SPAM]` string, or body of which contains the word `spam`.

Note that if parameters of the utility contain a search criterion and a list of file identifiers, the search will be performed within these files (the search conditions are combined in a conjunction - `AND`).

```
drweb-qcontrol --stat --search-from ai@5 def/backup/%
```

outputs statistics on all archived messages to the console (the messages which identifier corresponds to the specified `def/backup/%` template), sender of which is `ai@5`:

```
1. def/backup/5/00014F95.DW_SHOT_PRODUCT.1LXzg1  
from: ai@5; to: to@drweb.com; time:  
2008-8-14 15:1:46
```



drweb-lookup: Lookup Validation

The `drweb-lookup` utility allows validation of [Lookup](#) search results specified in the [Dr.Web MailD](#) settings.

The utility is launched with the following command:

```
drweb-lookup [parameters] <query>
```

where `<query>` – different types of `Lookup` used for the search and `[parameters]` – command line parameters.

The following command line parameters are available:

Short case	Extended case	Arguments
-h	--help	
<u>Description:</u> Output short help information on the command line parameters to the console and exit		
-v	--version	
<u>Description:</u> Output information on the utility version and exit		
-l	--level	<verbosity level>
<u>Description:</u> Set the logging verbosity level . The following levels are available: Quiet, Error, Alert, Info, Debug		
-i	--ipc-level	<verbosity level>
<u>Description:</u> Set the verbosity level for IPC records (interaction with <code>drweb-maild</code>). The following levels are available: Quiet, Error, Alert, Info, Debug.		
	--syslogfacility	<syslog flag>
<u>Description:</u> Set the subsystem type , used by syslog service for message output (if this service is used for logging, see the next parameter description). The following types are available: Daemon, Mail, Local0, ..., Local7.		
	--log-filename	<file name>
<u>Description:</u> Set the name of the log file or <code>syslog</code> (if the syslog service is to be used for logging).		
-a	--agent	<file path>
<u>Description:</u> Set the path to the Dr.Web Agent socket to receive configuration (by default, if the parameter is not specified, the <code>%var_dir%/ipc/.agent</code> path is used). If the switch is specified without a path, configuration from Dr.Web Agent is not requested.		
-t	--timeout	<time period>
<u>Description:</u> Set the maximum allowed time to wait for response from Dr.Web Agent when requesting configuration.		
-q	--query	<searched string>
<u>Description:</u> String that is a search object. If the "-" is specified, the utility reads a search value from the standard input.		
-e	--exist	
<u>Description:</u> Specify that only check for the searched object in the <code>Lookup</code> is required, without getting the value (the output can be one of the following: FOUND or NOT FOUND depending on the query result).		

**Examples:**

```
drweb-lookup -q q -e e,w
q NOT FOUND
```

```
drweb-lookup -q q -e q,q
FOUND q
```

```
drweb-lookup -q test@drweb.com -e 'ldap:///?displayName?sub?(mail=$s) '
FOUND test@drweb.com
```

```
drweb-lookup -q test@drweb.com 'ldap:///?displayName?sub?(mail=$s) '
notify.virus=block, notify.virus=allow(rcpt), drweb/ProcessingErrors = pass
```

```
drweb-lookup -q test@drweb.com "odbc:select rules from maild where a='\$s'"
scan = all:-drweb
```

drweb-inject: Sending Mail

The **drweb-inject** utility allows delivery of local mail via **Sender**. The utility receives a message body from standard input and returns 0 for success and not 0, if an error occurs. The sent email message can be piped to the utility via `|` conveyor (for example, as a `cat` command result) or redirected to the standard output via `<`.

The following command line parameters are available:

Short case	Extended case	Arguments
	<code>--help</code>	
<u>Description:</u> Output short help information on the command line parameters to the console and exit		
	<code>--version</code>	
<u>Description:</u> Outputs information on the utility version and exit		
	<code>--agent</code>	<code><file path></code>
<u>Description:</u> Set the path to the socket of Dr.Web Agent component to receive configuration (by default, if the parameter is not specified, the <code>%var_dir%/ipc/.agent</code> path is used). If the switch is specified without a path, configuration from Dr.Web Agent is not requested.		
	<code>--timeout</code>	<code><time period></code>
<u>Description:</u> Set the maximum allowed time to wait for response from Dr.Web Agent when requesting configuration.		
	<code>--id</code>	<code><identifier></code>
<u>Description:</u> Set the unique identifier of Sender used for mail dispatch. If not specified, the default Sender is used.		
<code>-f</code>	<code>--env-from</code>	<code><address></code>
<u>Description:</u> Set the message sender (for the message envelope). If not specified, the parameter value is the name of the user under whose account the utility is launched. If the username is not found, the utility exits with a non-zero error code.		



Short case	Extended case	Arguments
-F	--from	<address>
<u>Description:</u> Specify the From field value if the despatched email message does not have that field.		
-i	--ignore-dot	
<u>Description:</u> Instruct not to treat a string with the only dot character (".") as a terminating symbol of message entry.		
-t	--extract-recipients	
<u>Description:</u> Instruct to add all message recipients from the To field to the recipients in the email envelope.		

Example of mail dispatch with the use of the `drweb-inject` utility:

```
cat /var/drweb/msgs/out/failed/00000A59tNvGZ8  
| drweb-inject -f sender@domain rcpt@domain
```

This command instructs to send a message, saved to the `00000A59tNvGZ8` file and located in the `msgs/out/failed` directory. This directory stores all messages that failed to be sent including deferred ones. As an `-f` parameter value, the sender is specified and then - the recipient.

Note that when a message is sent from the database, **Dr.Web MailD** uses message files instead of envelope files (the latter have the same ID, but the extension is `.envelope`)!

To extract the list of recipients from the message, the `-t` parameter is used (in this case, specify only the message sender):

```
cat absGRjJ0to1Ubye |  
drweb-inject -t -f sender@domain
```

Message Processing Rules

Purpose of Message processing rules

Message processing Rules allow adjustment of settings used for message processing and message delivery depending on conditions. Each condition checks some of message parameters such as sender and recipient addresses, names of viruses or other threats detected in the message, its size and others. You can also specify different combinations of settings check and thus change the check procedure.

You can specify general Rules for all processed messages as well as Rules connected with certain users, their aliases and groups.

General Rules of message processing are set in the **Dr.Web MailD** configuration file, in the `[Rules]` [section](#). Rules of message processing connected with users and user groups are set in the [built-in database](#).

Storing rules in the built-in database is rational if the number of users with individual settings is great and, therefore, processing of Rules in the configuration file becomes inefficient as complexity of rule search is proportional to the number of rules specified in the file. In this case, Rule search in the database is more efficient and, moreover, optimizes memory usage.

Order of message processing and Rule search

When processing a message with a plug-in or another component, it can request a parameter value from **MailD core**. In this case, the required parameter value is selected according to the following algorithm:

- View of Rules in the built-in database that are connected with the message recipient (specified in



the RCPT TO).

- View of Rules in the built-in database that are connected with all user groups of the recipient. The view is performed in the reverse order: starting from settings of the last user group to the first user group in the list.
- View of Rules specified in the [Rules] section.

Note the procedure of Rule search

- All Rules in the current group are checked in the same order as they are set.
- For each rule, `CONDITION` is checked - if the conditional part is true, the required parameter value is searched in the `SETTINGS` part of the rule.
- If `CONDITION` is false, the required value is searched in the next Rule.
- If `CONDITION` is true and is followed by a `cont` directive, search continues in the next Rule. If the true `CONDITION` is followed by a `stop` directive, rule search stops regardless whether the required value is found or not.

Parameter value is determined according to the rule search results in the following way:

- If the searched parameter is found in one of the matching rules, the value is taken from the `SETTINGS` part (note that if more than one of the matching Rules contain this parameter, its result value depends on the parameter semantics. For details, see [Rule Format](#) and [Special Cases](#)).
- If no rule is found, no rule is matching or no matching rule contains the searched parameter, its value is taken from the corresponding section of the configuration file.
- If the searched parameter is not specified in the configuration file, the default parameter value is taken.

Rule Format

Rule format

Each message processing rule is specified as a string of the following type:

```
CONDITION stop|cont [SETTINGS]
```

where:

- **CONDITION** – condition which must be true for a message to which settings and/or actions specified in the **SETTINGS** part are applied.
- **SETTINGS** – settings and/or actions applied to a message for which **CONDITION** part of the rule is true.

Note that the **SETTINGS** part can be absent. That can be in one of the following cases:

1. If the Rule does not contain specific settings and is used for filtering;
 2. If the Rule uses settings that are loaded from an external source while the **CONDITION** is checked ([Lookup](#) from LDAP, DB, file or another source is used).
- Directive following the **CONDITION** part, specifies actions to be performed if the **CONDITION** is true for a message:
 - **stop** – stop further search of matching Rules;
 - **cont** – continue search Rules downwards.

If **CONDITION** is false for a message, the directive specified after that has no effect: Rule search continues.



If a rule is too long and cannot be specified on a single line, insert a "\" character at the end of the line and continue the rule on the next line.



Rule CONDITIONS

Each of the `CONDITION` parts is a combination of Boolean terms:

```
BOOL_TERM [<log_op> BOOL_TERM]
```

where `BOOL_TERM` – Boolean term that can get either true or false value after the message parameter is checked (the parameter can be set with the `[param_name:][value]` expression) or identically equal to `true` or `false`. A Boolean term has the following format:

```
<[param_name:][value]> | true | false
```

`param_name` – parameter name, `value` – parameter value.



Do not use white spaces between the parameter name and its value (that is, after the colon character) in Boolean terms.

Names of parameters that can be used in rule conditions are presented in the following table:

Parameter name	Description	Value type
any	Either sender's or recipient's address. Example: <code>any:regex:*@domain.com</code> Either sender's or recipient's address must be in the <code>domain.com</code> .	Lookup
from, sender	Address of the message sender. Example: <code>from:admin@domain.com</code> Sender of the message must have the <code>admin@domain.com</code> address.	Lookup
to, rcpt	Address of the message recipient. Example: <code>"rcpt:ldap:///??sub?(mail=\$s)"</code> All message recipients must be found in the LDAP by the mail field	Lookup
block	Object that caused a plug-in to block the message. A message is considered blocked if <code>reject action</code> was applied to it. In this case, the plug-in that blocked the message returns a list of strings describing the reasons for blocking (there can be more than one reason). For example, Drweb plug-in , in the event of virus (or another known threat) detection, returns the threat name. If the message was blocked for a different reason (for example, due to <code>SkipObject event</code>), the plug-in returns a value of the <code><parameter> = <value></code> configuration string (in the format <code>"<parameter>: <value>"</code>), execution of which caused the block. For example, HeadersFilter plug-in can inform that a message was blocked due to the missing header <code><header></code> specified in the MissingHeader parameter . If so, the plug-in response is as follows: <code>"MissingHeader: <header>"</code> . Examples: 1) <code>block:file:viruses.txt</code> Name of the object or reason that caused the message block; must be specified in the list from the file (it is implied to be the name of the detected threat). 2) <code>"block:regex:.*skip.*"</code> Description of a reason that caused the message block; must contain the <code>"skip"</code> substring (for example, action applied upon <code>SkipObject event</code>).	Lookup
client-ip	IP address of the message sender (if receiving information on the	Lookup



Parameter name	Description	Value type
	sender's IP address was specified in the Maid core settings). Note that you cannot use CIDR format here. Example: <code>client-ip:127.0.0.1</code> The message must be sent from the local machine.	
<code>client-port</code>	Port of the client that sent the message. Example: <code>client-port:1234</code> The message must be sent from the 1234 port.	Port number
<code>server-unix-socket</code>	Absolute path to the Unix socket file used for receiving the connection. Example: <code>server-unix-socket:/var/drweb/ipc/sock1</code>	Path to UNIX socket
<code>server-ip</code>	IP address of the interface used by Receiver to get the message. Note that you cannot use CIDR format here.	Lookup
<code>server-port</code>	Port of the server which accepted the connection.	Port number
<code>id</code>	Unique identifier of Receiver that got the message (if the identifier was set in the Receiver settings).	Identifier string
<code>auth</code>	Indicates whether the client which sent the message obtained authorization. Note that the parameter does not have an argument. Example: <code>auth:</code>	Absent
<code>size</code>	Size of the message. Specify a comparison operator before the size: <ul style="list-style-type: none">• <code>!=</code> – Not equal.• <code>==</code> – Equal.• <code><</code> – Less than.• <code>></code> – Greater than.• <code><=</code> – Less or equal.• <code>>=</code> – Greater or equal. If a comparison operator is not specified, the default <code><=</code> operator is used. Size is specified in KB, MB or GB (enter the corresponding suffix after the number - k, m, g). Example: <code>"size:>=10m"</code>	Size or comparison operator
<code>score</code>	Message score. Before the size, it is required to specify the comparison operator. If not specified, the default <code><=</code> operator is used. Example: <code>"score:!=1000"</code>	Number or comparison operator

Note that:

- If the parameter name is not specified in the Boolean term, the **any** parameter is used by default (for example, `user@domain.com stop <some_settings>` is equal to the following Rule: `any:user@domain.com stop <some_settings>`).
- If the parameter value contains white spaces or `'|', '&', ')', '(' '!' '=' ,'` characters, the value must



be quoted. To use a `'` character within the quotation marks, escape the character with the backslash `'\'`.

- If the Rule contains an empty address, specify it as follows `''`, and do not use angle brackets (`<>`) for that. For example, `from:'' stop scan=no`.

To create composite conditions, use the following logical operators to join simple Boolean terms: `AND` (`&&`), `OR` (`||`), `NOT` (`!`). You can also use brackets to set priority of logical operations. Condition must contain at least one Boolean term.

Example 1:

```
true stop <some_settings>
```

Settings `<some_settings>` are applied unconditionally if the Rule is used during the processing (that is, the condition is true for any message). As the `stop` directive is specified afterwards, all subsequent rules are not used.

Example 2:

```
sender:test && "size:>=10k" cont scan=no
```

This condition is true if the message sender is `"test"` and the message size is greater than 10 KB. In this case, the message is passed without further check. Note the usage of quotation marks: they are necessary in this example.

Example 3:

```
!("rcpt:ldap:///??sub?(mail=$s)" OR auth:) stop
```

This Rule is applied if at least one of the recipients, specified in `TO:`, is not found in LDAP by `mail` field and the sender is not authorized.

Rule SETTINGS

SETTINGS part of a rule is a sequence of `Parameter = Value` pairs:

```
[plugin_name/]param = value,  
[plugin_name/]param = value ...
```

where `plugin_name` (if specified) – name of the [plug-in](#), which parameter values are used, `param` – parameter name, `value` – parameter value.

If the plug-in name is not specified, the rule configures parameters of the main **Dr.Web MailD** configuration file (section name is not specified). For example, `AdminMail=root@domain` directive means that the value of the `AdminMail` parameter in the [Notifier] [section](#) of **Dr.Web MailD** configuration file is changed.

At least one `Parameter = Value` pair is required. List items are separated by commas, so if `value` contains a comma, enter a backslash `"\"` before the character to escape it.

Example 1:

```
sender:a@drweb.com cont headersfilter/Action = pass, vaderetro/max_size = 100k
```

When this rule is applied (to the `a@drweb.com` sender), `Action = pass` parameter value is used for **Dr.Web HeadersFilter** [plug-in](#) and the maximum size (`max_size`) of the message checked by **Vaderetro** plug-in is set to `100k`. After that, search of matching subsequent Rules continues (as the `cont` directive is specified).

**Example 2:**

```
to:a@drweb.com cont drweb/ProcessingErrors = pass\, redirect(err@drweb.com)
```

Note that one **ProcessingErrors** parameter for **Drweb plug-in** is set, and this parameter has two values separated by commas (`pass, redirect(err@drweb.com)`). Thus, it cannot be enclosed in quotation marks as a configuration parser treats it as one value and does not split them into substrings when parsing the **ProcessingErrors** parameter. The comma is escaped.

Rule processing is performed in the following order: up to down and left to right, so:

- If the value of the same parameter in one Rule (**SETTINGS** part) is specified several times, the successive parameter value substitutes the previous one.
- If several Rules are matching, where different values are specified for the same parameter (in the **SETTINGS** part), the first assigned value is set and the subsequent values are ignored.

Example 3 (one Rule with several values for the same parameter):

```
true cont html=yes, html=no
```

Value of the **html** parameter (the parameter description is provided below) is set to `no`.

Example 4 (several matching rules with different parameter values):

```
true cont html=yes
true cont html=no
```

Value of the **html** parameter is set to `yes`.

Behaviour described above is relevant for all parameters except for those with **additive semantics**. A subsequent value for such parameter does not conflict with the previous one, but is added to it. As a result, all found parameter values are joined in a single list of values. At that, when a **stop** directive is specified, Rule search stops and the values gathered by that moment are assigned to the parameter. In the present document, parameters with additive semantics are marked with the **A** icon in their descriptions.

Moreover, some parameters are of the **cloning** type. These parameters are processed for every receiver separately: that is, if different values of the parameter are specified for different message recipients, a separate copy of the message is created for each of them and corresponding settings are applied to each copy. In the present document, parameters that support cloning are marked with the **C** icon in the description.

Parameters used in Rule SETTINGS

Parameters used in the **SETTINGS** part of a Rule, are divided into the following categories:

- parameters used in Rules only.
- parameters which values are specified in the **Dr.Web MailD** configuration file or configuration files of other modules or plug-ins.

In the present document, parameters that can be used in Rules, are marked with **R** icon.

1. Parameters used only in Rules:

```
html =
{logical}
```



Enables or disables generation of notifications in HTML format.

If the value is set to `Yes`, **Dr.Web MailD** generates notifications in HTML format; otherwise, they are generated as text.

Note that this parameter does not manage DSN.



```
quarantine =  
{logical}
```

Enables or disables moving rejected messages to **Quarantine**.

If the value is set to **Yes**, such messages are quarantined.

Otherwise, messages cannot be move to **Quarantine**, even if they were rejected.

```
scan =  
{text}
```



Determines which **plug-ins**, specified in the [Filters] [section](#), are used by **Dr.Web MailD** to check messages.

If the parameter value is set to **All**, **Dr.Web MailD** uses all plug-ins. If the parameter value is set to **No**, **Dr.Web MailD** uses no plug-ins. Plug-ins are listed with a colon ":" as a delimiter. To exclude a plug-in from the list, use a minus sign "-" with no white spaces before the plug-in name. Note that if the value is set to **All**, plug-in names cannot be specified in the list without a minus sign, as that is useless.

Examples:

scan = All - use all plug-ins

scan = no - do not use any plug-in

scan = All:-plugin1 - use all plug-ins except for plugin1

scan = Plugin1:Plugin2 - use only Plugin1 and Plugin2 plug-ins

scan = All:Plugin1 - invalid, as it is not allowed to use plug-in names without "-" after All

scan = -Plugin1:All - invalid as All must be specified first

scan = -Plugin1 - invalid as All is missing before the plug-in name.



```
notify[.<notification type>] =  
{allow | block}  
[(<address types>)]
```

This parameter configures [different types](#) of notifications.

If the value is set to `allow`, sending notifications of the corresponding type is allowed. If the value is set to `block`, sending notifications of the corresponding type is blocked. If the notification type is not specified, this parameter value is applied to all notifications.

What notification types are available depends on what types are supported by [Dr.Web Notifier](#). Additionally installed plug-ins can add their own notification types.

By default, the following notifications are available:

- **notify.Virus** - notifications on detection of a virus in a message
- **notify.Cured** - notifications on cured viruses detected in a message
- **notify.Skip** - notification on skipped messages
- **notify.Archive** - notifications on messages that were not checked due to archive check restrictions
- **notify.Error** - notifications on errors that occurred during message check
- **notify.Rule** - notification on a message blocked by a Rule
- **notify.License** - notifications on messages that were not checked due to licence restrictions
- **notify.Malware** - notifications on detected malicious programs.

The parameter value can be followed by an optional modifier indicating what types of recipient's addresses match this parameter. You can specify several address types, separated by colons. The following modifier values (recipient's address types) are available:

- `sender` - notifications sent to the message sender
- `rcpt` - notifications sent to the message recipients
- `admin` - notifications sent to the administrator
- `any`, or a modifier is not specified - notifications sent to the addresses of any type.

Examples:

`notify=block` or `notify=block (any)` - block notifications of all types.

`notify.Virus=block (sender:admin)` - block notifications sent to the administrator and message sender upon a virus detection.

If for a notification of a certain type (`<type>`) no `notify.<type>` rule is found, it is assumed that the corresponding notification is disabled.

Note that this parameter does not manage DSN.



```
NotificationNamesMap =  
name1 file_name1,  
name2 file_name2,  
...
```

Allows mapping of used template names to the new ones. For example, the parameter can be used to generate notifications of different types depending on the envelope.

As the parameter value, the following information is specified:

- nameN - name of the requested notification, for which a new template file is set. For the list of names, see the **notify** parameter description. Besides that, you can also specify **report** for general report templates and **dsn** - for DSN.
- file_nameN - part of a new template file name. The full name is composed according to the [following scheme](#): file name is prepended with the respective prefix **sender_**, **rcpts_**, **admin_**, **report_** or **dsn**, file extension is substituted with **.msg**. The resulting file name, e.g. **sender_file_nameN.msg**, is searched in the directory specified in the **TemplatesBaseDir** parameter value in the [Notifier] [section](#).

Example:

```
NotificationNamesMap = virus my-virus, archive  
my-arch
```

```
SenderAddress =  
{address1|address2|...}
```

Address where the message must be sent. The address is transmitted to **Sender**.

It is possible to specify several address separating them by the "|" character (similar to the **Router** parameter from the [Sender] [section](#)).

When using the **SenderAddress** parameter in Rules of the following type:

```
<CONDITION> cont SenderAddress =  
address1|address2|address3
```

a message that satisfies <CONDITION> is sent to the first available address from the list. That is, if **address1** is unavailable, the message is sent to **address2**, and if that address is also unavailable - to **address3**.

This parameter can use the following special macros:

- CLIENT-IP - IP address of the client which sent the message
- CLIENT-PORT - Port of the client from which sent the message
- SERVER-IP - IP address served by **Dr.Web MailD**
- SERVER-PORT - Port on which the message was received by **Receiver**.

If **Sender** supports this parameter, the component transmits the message to the specified address.

For example, the following rule

```
true cont SenderAddress=inet:10025@CLIENT-IP
```

redirects all incoming messages to the port 10025 of the host which sent the message.

In the current **Dr.Web MailD** version, this parameter is supported only by **drweb-sender** [module](#) with SMTP/LMTP send method.



Default parameter values, listed in the table above, are set in the [Rule] configuration file [section](#) for which the name is not specified.

To ensure proper functionality of the CLIENT-IP and CLIENT-PORT macros, used in the **SenderAddress** parameter, specify the following parameter values:

- In the [Receiver] [section](#): **RealClients** = yes
- In the [Maild] [section](#): **GetIpFromReceivedHeader** = yes

2. Parameters from the main configuration file which can be used in Rules:

Section	Parameters
[Maild]	RedirectMail MaxScore MaxScoreAction LicenseLimit EmptyFrom ProcessingErrors UseCustomReply ReplyEmptyFrom ReplyProcessingError ReplyMaxScore
[Notifier]	C AdminMail FilterMail C NotifyLangs
[Filters]	C <plug-in>/use - This parameter allows or restricts using of the specified plug-in when checking messages. It is of Logic type {yes, no}. Note that this parameter can be used only in Rules and is not specified in the configuration file! C <plug-in>/max_size <plug-in>/log_level <plug-in>/log_ipc_level <plug-in>/syslog_facility <plug-in>/path_to_lib

3. Plug-in parameters that can be used in Rules:

Plug-in	Parameters
Drweb	HeuristicAnalysis AddXHeaders Paranoid A RegexsForCheckedFilename LicenseLimit Infected Suspicious Incurable Adware Dialers Jokes Riskware Hacktools SkipObject ArchiveRestriction ScanningErrors ProcessingErrors BlockByFilename



Plug-in	Parameters
	<ul style="list-style-type: none">UseCustomReplyReportMaxSizeReplyInfectedReplyMalwareReplySuspiciousReplySkipObjectReplyArchiveRestrictionReplyErrorReplyBlockByFilename
<u>Vaderetro</u>	<ul style="list-style-type: none">FullCheckNoHamFromAddVersionHeaderAddXDrwebSpamStateNumHeaderAddXSpamLevelAddXHeadersCheckDeliverySubjectPrefixNotifySubjectPrefixUnconditionalSpamThresholdUnconditionalSubjectPrefixSpamThresholdUnconditionalActionActionNotifyActionSpamCustomReply WhiteList BlackListCheckForVirusesAllowRussianAllowCJKUseCustomReplyFromProtectedNetworkScoreAddProtectedNetworkReplyCacheLifeTimeReplyToProtectedNetworkScoreAdd
<u>HeadersFilter</u>	<ul style="list-style-type: none">ScanEncodedHeaders RejectCondition AcceptConditionFilterParts RejectPartCondition AcceptPartCondition MissingHeaderActionUseCustomReplyReplyRuleFilter
<u>Modifier</u>	<ul style="list-style-type: none">EncodingUseCustomReplyReplyRuleFilter LocalRules - Local rules of message check with the plug-in (similar to the format of the GlobalRules plug-in parameter). Note that this parameter can be used only in Rules and is not specified in the plug-in configuration file! For the plug-in rule format, see



Plug-in	Parameters
	the plug-in description .

Note that the operational logic of the `<plug-in>/use` parameter is similar to the one of the `scan` parameter (exclude a plug-in from or include it in the list of the used plug-ins). The parameter can be also used to include or exclude plug-ins on Rule matching, as the `scan` parameter does not have additive semantics.

Example:

If it is required to disable **Drweb** and **Vaderetro** plug-ins with the use of two rules, then the following two Rules

```
to:regex:test@.* cont scan=all:-vaderetro
to:regex:test@.* cont scan=all:-drweb
```

result in disabling only **Vaderetro** plug-in (as repeated change of the `scan` parameter is rejected). However, if the second Rule is changed as follows:

```
to:regex:test@.* cont scan=all:-vaderetro
to:regex:test@.* cont drweb/use=no
```

Vaderetro plug-in will be disabled after appliance of the first Rule and **Drweb** plug-in will be disabled after appliance of the second one. The same effect can be reached if `scan=all:-vaderetro` is substituted to `vaderetro/use=no` in the **SETTINGS** part of the first Rule.

In the **SETTINGS** part of a Rule, you can use parameter values set in a named settings group (specified as a `[Rule:<section name>]` section in the main configuration file). For that purpose, specify the `rule=<group name>` directive. In the current **Dr.Web MailD** version, no more than one `rule` parameter can be used (for examples, refer to `[Rule]` [section description](#)).

Special Cases

Processing messages with several recipients

If a message has different recipients and Rules set different parameter values for them (for example, a rule setting `html=no` matches one recipient, and a rule setting `html=yes` matches another), the conflict is resolved in the following way:

1. If the parameters, which have different values specified for different recipients, allow cloning, the message is cloned (separate message copies are created and sent to the corresponding recipient). Each copy has the parameter value set for its recipient.
2. If the parameter does not allow message cloning, different parameter values for different recipients are substituted with the same parameter value from the corresponding section of configuration file settings. If the parameter is not specified in the configuration file, the default value is used.
3. If the same rule matches all the recipients or in the matching rules the parameter value is the same, this value is used.

Note that if parameter values are not changed in the Rules, values for these parameters are taken from the corresponding configuration file. For those parameters, values of which are not specified in the configuration file, default values are used. The message is sent for check to all plug-ins, specified in the `[Filters]` section, except for those indicated in Rules as not used (`<plug-in>/use = no` or `scan=all:-<plug-in>`).

**Example 1:**

a) The following two rules are specified:

```
[Rules]
to:user1@domain.ru cont drweb/Suspicious = pass\, quarantine\, notify
to:user2@domaun.ru cont drweb/Suspicious = discard\, quarantine\, notify
```

b) Message with the following headers is processed:

```
FROM: <another_user@externaldomain.com>
TO: <user1@domain.ru>, <user2@domain.ru>
```

In this case, the message is not cloned, as setting of **Drweb plug-in** specified for **Suspicious** is taken from the `plugin_drweb.conf` configuration file. This is because settings for recipients are different and the `drweb/Suspicious` parameter is not of the cloning type.

Example 2:

a) The following three rules are set:

```
[Rules]
to:user1@domain.ru cont drweb/Suspicious = pass\, quarantine\, notify
to:user2@domain.ru cont drweb/Suspicious = discard\, quarantine\, notify
from:another_user@externaldomain.com cont drweb/Suspicious = reject\, add-
header (BLA:BLA)
```

b) The processed message is the same as in the previous example:

```
FROM: <another_user@externaldomain.com>
TO: <user1@domain.ru>, <user2@domain.ru>
```

Only the last rule is applied to the message, as `drweb/Suspicious` does not have additive semantics and is not of the cloning type. Thus, different settings specified in the first two rules cause the value to reset to the default. The third rule is also applied to the message and sets the `reject,add-header (BLA:BLA)` value to the `drweb/Suspicious` parameter.

Example 3:

a) The following Rules are set:

```
[Rules]
to:user1@domain.ru cont NotifyLangs=ja, AdminMail=admin2@domain.ru
to:user2@domain.ru cont NotifyLangs=ja, AdminMail=admin2@domain.ru
to:user3@domain.ru cont NotifyLangs=ja, AdminMail=admin2@domain.ru
from:root@domain.ru cont NotifyLangs=ru
to:admin@domain.ru cont NotifyLangs=ru\, ja
```

b) Default **settings for MailD notifications sending** are:

```
...
[Notifier]
...
AdminMail = admin@domain.ru
FilterMail = drweb@domain.ru
NotifyLangs = en
...
```



c) The following infected message is processed:

```
FROM: <root@domain.ru>
TO: <user1@domain.ru>, <user2@domain.ru>, <user3@domain.ru>
```

As all notifications are sent from the address specified in **FilterMail**, the following 5 notifications are sent:

- o From <drweb@domain.ru> to the administrator <admin@domain.ru>;
- o From <drweb@domain.ru> to the sender <root@domain.ru>;
- o From <drweb@domain.ru> to the recipient <user1@domain.ru>;
- o From <drweb@domain.ru> to the recipient <user2@domain.ru>;
- o From <drweb@domain.ru> to the recipient <user3@domain.ru>.

Administrator with the <admin@domain.ru> address receives notifications in Russian and Japanese languages, sender with the <root@domain.ru> address - in English language, and all users (<user1@domain.ru>, <user2@domain.ru>, <user3@domain.ru>) - in Japanese language.

Note that the `from:root@domain.ru cont NotifyLangs=ru` rule is not applied in this example as when sending notifications to <root@domain.ru>, the address is treated as the notification recipient's address, so notifications are generated with the use of the default [language file](#) (in English language). If it is required to use Russian language for notifications, specify the following Rule:

```
to:root@domain.ru cont NotifyLangs=ru
```

Rules with implicit SETTINGS part

As [mentioned above](#), **SETTINGS** part of a rule can be absent. If so, parameters are requested directly from the server with the use of [Lookup](#) specified in the **CONDITION** part. It can be useful, for example, when working with LDAP or databases.

Example:

```
to:regex:.*@drweb.com && "ldap:///?drwebRules?sub?(mail=$s)" cont
```

In this Rule, if a recipient is in the `drweb.com` domain and sender or all the recipients satisfy the LDAP condition `mail=$s`, parameters in the **SETTINGS** part are substituted from the `drwebRules` field of the LDAP request. Values are substituted for every new message and then are cached while the message is checked - thus, a user can change settings in "hot" mode without server restart. Note that `Lookup` to LDAP is enclosed in quotation marks because of the brackets used in the `Lookup`. Fields returned by `drwebRules`, must be correct **SETTINGS** strings (that is `<parameter>=<value>[, <parameter>=<value>, ...]`).

For example, if stored in a database table, they must be of the following type:

Example:

Address	Rules
test1@drweb.com	VadeRetro/SubjectPrefix = \"spam\",modifier/localrules=select message\,append_text "Some Text"
test2@drweb.com	headersfilter/MissingHeader = Date,headersfilter/MissingHeader = From, headersfilter/MissingHeader = To

Also note that if **SETTINGS** part is specified in the Rule and **CONDITIONS** part contains [Lookup](#) that retrieves settings from a data source, these settings are to be joined (concatenated) to the **SETTINGS on the left** (first, settings retrieved from the source are specified and then those which are set directly in the **SETTINGS** part of a Rule). It is important to note in case of a conflict between different values of the same parameter inserted into the Rule from different sources.

**Example:**

a) Let us assume that a `table` in the database, access to which is [configured](#) using ODBC, is of the following type:

Address	Rules
test1@drweb.com	modifier/LocalRules = select message\, append_text "text from DB"

b) In the [configuration file](#), the following Rule is specified:

```
"to:odbc:select Rules from table where Address='$s'" cont modifier/LocalRules =
select message\, append_text "text from rule", modifier/LocalRules = quarantine
```

In this case, when the Rule is matching, the following `SETTINGS` are applied for the message sent to test1@drweb.com:

```
modifier/LocalRules = select message, append_text "text from DB", select message,
append_text "text from rule", quarantine
```

Note that the `modifier/LocalRules` parameter has additive semantics and new values do not reset the previous ones but are added separated by commas.

If the message has other recipients, except test1@drweb.com, for which other `modifier/LocalRules` values are specified in the database (or the values are not specified at all), all `modifier/LocalRules` values from the database are ignored for all of the recipients. Only the settings set in the Rule are used (the same value for all the recipients):

```
select message, append_text "text from rule", quarantine
```

When necessary, you can override `OnError` and `SkipDomains` settings of the used data source.

Example:

```
to:ldap:onerror=exception|skipdomains=file:/etc/drweb/skipdomains.list|
///?description?sub?(cn=$u) cont
```

According to the specified condition, all names of the message recipients (username part in the username@domain address), according to the used `$u` macro, must be taken from LDAP. However, `SkipDomains` setting is also specified in the `Lookup` and instructs to skip domains without request (domain part in the address) that are found in the `/etc/drweb/skipdomains.list` file. Thus, the rule is processed as follows:

1. List of recipients is retrieved from the message (field To:).
2. Address of another recipient is taken.
3. Domain membership to the list of domains to be skipped without request is verified. If so, the next address in the list is checked (step 2). If not, LDAP request is executed. Note that if the message is sent to one recipient, and the domain is in the list of skipped domains, this rule forms no settings for the message.
4. If the request result is positive, content of `description` field is used as a setting (concatenated on the left to the settings retrieved earlier), then step 2 is executed. Otherwise, if the request result is negative (user name is not found in the data source), the Rule condition is false for the message, this Rule is rejected and all settings retrieved from LDAP are discarded. Remaining requests to LDAP are not performed.
5. If LDAP request fails, that is considered as a message processing error. Processing of this message stops and, according to `OnError=exception` setting, action specified in the `ProcessingErrors` ([Maid] [section](#)) parameter is performed. For more information on how errors are handled in Rules, see below.

As the `cont` directive is specified and no errors occurred, viewing of other rules continues. Note that if



the message is sent to several recipients and different parameter values are retrieved for them from LDAP, the setting values are either determined after cloning of the message or substituted with parameter values from the configuration file (as described above in the [Processing messages with several recipients](#) paragraph).

Joining Rule settings from different data sources

As described [at the beginning](#) of this section, values of required parameters are retrieved from Rules stored in the built-in database, then - from Rules specified in the [Rules] section, and after that - from the configuration file. Let us consider an example which illustrates joining of settings retrieved from different sources according to the described algorithm.

Example:

a) Let us assume that the following Rules are specified for the `test@drweb` user in the [built-in database](#):

```
> email-info test@drweb.com
test@drweb.com A=1 S=1
  name:
  aliases: alias_test@drweb.com
  groups: divine good evil
  rules:
    1: true cont modifier/LocalRules = select message\, append_text
"Scanned! (1)",\
      modifier/LocalRules = quarantine
    2: true cont modifier/LocalRules = select message\, append_text
"Scanned! (2)"
    3: "rcpt:odbc:select rules from maild where addr='$s'" cont
  custom:
```

Three processing Rules relate to `test@drweb.com` address; at that, the third rule has implicit SETTINGS part that checks condition using `Lookup` and imports selected set (`rules` field value) as the SETTINGS part of the Rule. The Rule uses data from the `maild` database table; access to this table is also [configured](#) via ODBC.

b) Let us assume that the table entry which relates to `test@drweb.com` address is as follows:

addr	rules
test@drweb.com	modifier/LocalRules = select message\, append_text "Scanned! (3)"

c) And let us also assume that the following Rule is specified in the [Rules] section of the main configuration file:

```
[Rules]
true cont modifier/LocalRules = select message\, append_text \
"Scanned! (4)", modifier/LocalRules = quarantine
```

In this case, the following procedure is performed for the message with the `test@drweb.com` sender address:

- 1) At first, Rules from the built-in database are applied. As they are identically true (`true` is specified as condition), their values are used. Moreover, the `modifier/LocalRules` parameter has additive semantics and values from the Rules are sequentially concatenated.
- 2) According to the third Rule, `rules` field value for the `test@drweb.com` address is queried from the database. The received value is added to the current concatenated parameter value.
- 3) As `stop` directive did not occur and the `modifier/LocalRules` parameter has additive semantics, true Rules in the [Rules] section of the configuration file are searched. Found values of the `modifier/LocalRules` parameter are concatenated.



As the result, for the message sent to `test@drweb.com`, the `modifier/LocalRules` parameter has the following concatenated value:

```
modifier/LocalRules = select message, append_text "Scanned! (1)", quarantine, select
message, append_text "Scanned! (2)", select message, append_text "Scanned! (3)",
select message, append_text "Scanned! (4)", quarantine
```

Error Handling and Rule Validation

Error handling

If an error is found in the line with a Rule, the error is logged and the Rule is ignored. Moreover, when `Lookup` is used, the error is handled according to the `OnError` parameter value (set for the data source or [overridden directly](#) in the `Lookup`).

Note that `Lookup` values and values of certain variables are not handled immediately - they are parsed when they must be used. Thus, errors in these elements can be detected only during message processing (when a Rule with an error is ignored).

Moreover, if during rule analysis a `Lookup` expression returns incorrect results, all the results are rejected and ignored (even if some of the results are correct).

Rule validation

To validate Rules, run `drweb-maild` [module](#) with special command line options that specify different properties of a message and the plug-in outputs to console all the settings from the Rule that are to be applied to the message. Available parameters are listed in the [Command line parameters](#) section (the **Module Specific Parameters** subsection).

Example of a Rule validation command:

```
$ ./drweb-maild --auth
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG notify* :
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG all : block
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG archive :
from=allow; admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG cured : from=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG error : from=allow;
admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG license :
admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG malware :
from=allow; to=allow; admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG rule : admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG skip : from=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG virus : from=allow;
to=allow; admin=allow;
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG scan : all
Thu May 29 16:03:44 2009 [3081324208] maild.rules DEBUG html : 1
```

In this example, processing of a message with a flag of successful sender authorization (`auth`) is emulated. Other message parameters are not specified. In the output, you can view all the settings that are applied to the message by the matching Rules (this example shows users of what type will receive notifications in each of the message check result, the message is checked with all connected plug-ins and reports are generated in the HTML format).



Unified Score

Unified Score technology allows detection of unwanted mail messages by the unified score assigned to each message. Message score is a signed integer. The greater it is, the higher is the probability that the message is unwanted, and vice versa - the smaller the number, the lower the probability that the message is unwanted. By default, a message is not considered spam if its score is less than the `SpamThreshold` parameter value (that is, 99 and less). If a message score is greater than the `SpamThreshold` value, but less than `UnconditionalSpamThreshold` (that is, from 100 to 999 by default), this message is considered spam. If a message score is greater than the `UnconditionalSpamThreshold` value (1000 by default), this message is considered to be an unconditional spam.

Message score can be modified in the following ways:

- in the parameters of the **Action** type you can use an optional [action](#) - `score (SCORE)`, where `SCORE` is an integer, which can be added to the current message score (or subtracted from the score if this value is negative).
- **Vaderetro anti-spam plug-in** assigns a score to the message, and this score is added to the total message score which is compared to spam threshold values).
- you can also modify a message score using Rules of some [plug-ins](#), as well as by changing some parameter [restrictions](#) in the `[Receiver]` [section](#).
- using **Reputation IP Filter** that allows modification of all message scores in the current session.

Message score is used in:

- **Vaderetro plug-in** that compares the score to the spam thresholds;
- conditions of [message processing rules](#) (with the `score` prefix);
- conditions of **Dr.Web Modifier plug-in** (using `add_score` и `set_score` commands);
- if a message score becomes greater than the `MaxScore` parameter value from the `[MailD]` [section](#) of the **Dr.Web MailD** configuration file, the message check is aborted and [action](#) specified in the `MaxScoreAction` parameter is applied;
- some [restrictions](#) that allow to apply different actions to the message depending on its current score;
- `drweb-receiver` to block whole sessions if total message score exceeds the `MaxSessionScore` parameter value from the `[Receiver]` [section](#).
- `score_filter` from **Reputation IP Filter** that allows filtration of IP addresses which have too large total score.

Reputation IP Filter

Reputation IP Filter is a technology for gathering statistics on each IP address connected to **Dr.Web for UNIX mail servers**. According to the statistics, **Reputation IP Filter** can either temporarily block an IP address or take other actions. This technology allows successful detection of spammers and resistance to DHA attacks.

Reputation IP Filter is enabled if either at least one filter is specified in the `ReputationIPFilter` setting (similar parameter in the `[Receiver]` [section](#)) or the `MaxConcurrentConnection` parameter value (in the same section) is not set to 0. By default, the `ReputationIPFilter` parameter value is set to `score_filter` and, therefore, an IP filter is enabled and IP addresses are filtered according to the average score assigned to all messages and sessions from these IP addresses (see below).

All information on IP addresses is stored in RAM and saved to files when the `drweb-receiver`



process terminates. File reading occurs only on `drweb-receiver` startup.

Files are saved and loaded only if at least one filter is specified in the `ReputationIPFilter` parameter. If no IP connection is established, information cannot be gathered and saved. Information on the connections is saved to `ipv4.bin` and `ipv6.bin` files (for IPv4 and IPv6 addresses respectively) to the directory specified in the `BaseDir` parameter from the [General] [section](#). If an error occurs during file saving or reading, the error is logged.



Data saved to these files is binary and depends on the operating system, thus, it is not recommended to use these files on another system.

Reputation IP Filter checks an IP address immediately after check of `SessionRestrictions` if the address is not marked as `trusted` (for details on `*Restrictions` and `trusted` flag, refer to the [description](#) of the [Receiver] section).

Thus, if it is necessary to prevent block of a certain IP address by **Reputation IP Filter**, mark this address as `trusted` in the `SessionRestrictions` parameter. If an IP address was mistakenly blocked by **Reputation IP Filter**, mark the address as `trusted` in the `SessionRestrictions` parameter, and all the subsequent connections from this IP will be ignored by **Reputation IP Filter**.

Reputation IP filter allows assigning a score to the IP address according to the gathered statistics on connections and temporary blocking of this IP address if its total score is greater than a threshold value.

The following filters are available: `anti_dha`, `errors_filter`, `score_filter`.

Reputation IP Filter checks an IP address immediately after `SessionRestriction` check is performed and only if the address is not marked as `trusted` (that is, if the address is marked as `trusted` during `SessionRestriction` check, the address is not processed by **Reputation IP Filter**).

Filters are listed using comma as a delimiter and checked in the order they are specified. For each filter, its name is specified first and then its parameters separated by spaces (all these parameters are optional).

Parameters are set as `NAME=VAL` pairs (there must be no white spaces between the value and equal sign).

General parameters for filters are described below (U - is a positive integer, I - is an integer, D - is a positive floating-point number):

- `min_msgs=U` - minimum number of messages passed for check to **MailID core**. After the value is exceeded, a corresponding filter is activated. If the value is set to 0, this parameter is ignored.
- `min_errors=U` - minimum number of errors registered on the stage of SMTP session. After the value is exceeded, a corresponding filter is activated. If the value is set to 0, this parameter is ignored.
- `min_wrong_rcpts=U` - minimum number of invalid recipients (declined after the `RCPT TO` command) transferred by the SMTP client. After the value is exceeded, the filter is activated. If the value is set to 0, this parameter is ignored.
- `min_conn=U` - minimum number of connections from the IP address. After the value is exceeded, the filter is activated. If the value is set to 0, then this parameter is ignored.
- `block_period=T` - sets blocking time for the IP address if it falls within the limits of the filter. T is of a {time} type. If the value is set to 0, IP address is not blocked even if it falls within the filter limits.
- `score=I` - a score to be assigned to all messages in the current session. Also it is added to the



general score of the IP address. If the score value is not equal to 0, this parameter will be applied instead of the `block_period` parameter, and a score will be assigned to the IP address instead of blocking.

Each of the filters has a set of unique parameters and specific set of default values for general parameters:

- `anti_dha` - resistance to DHA attacks (directory harvest attack). To use this filter, specify the full range of protected addresses as a value of the `ProtectedEmails` parameter in the [Receiver] [section](#).

Specific parameters:

- `wrong_per_valid_rcpts=D` - a ratio between the number of invalid message recipients (declined after the `RCPT TO` command) and the number of valid recipients. It is the main parameter defining filter operation. If no valid recipients are found, this value is considered equal to 1. If this parameter value is set to 0, the filter is totally ignored. Default value: 10.0

Default values for general parameters are:

- `min_msgs=0`
- `min_errors=0`
- `min_wrong_rcpts=20`
- `min_conn=0`
- `block_period=2h`
- `score=0`

- `errors_filter` - allows filtration of IP addresses according to the amount of errors that occurred during SMTP session established from the certain IP address.

Specific parameters:

- `errors_per_msg=D` - a ratio between the number of errors occurred during the SMTP session and the number of messages passed to **MailD core**. If no messages were passed, this number is considered equal to 1. If the parameter value is set to 0, this check is ignored. Default value: 0
- `errors_per_conn=D` - a ratio between the number of errors occurred during the SMTP session and the number of connections from this IP address. The filter is applied only when the parameter value is not 0 and at least one connection was established from this IP address. Default value: 2.0

If both parameters are specified, the `errors_per_msg` parameter is checked first, and the `errors_per_conn` parameter is checked after it. If values of both parameters are set to 0, the filter is ignored.

Default values for general parameters:

- `min_msgs=0`
- `min_errors=100`
- `min_wrong_rcpts=0`
- `min_conn=50`
- `block_period=2h`
- `score=0`

- `score_filter` - filters IP addresses according to the average score assigned to all messages and sessions from this IP address. It is included into the general **Unified Score** system and allows, for example, blocking of spammers on the stage when SMTP connection is established.

Specific parameters:

- `score_per_msg=D` - a ratio between the general score for the certain IP address (a sum of all scores of messages sent from the given IP address and scores of all sessions initiated from it) and the number of messages passed to **MailD core**. If no messages were passed, this



number is considered equal to 1. If the parameter value is set to 0, this check is ignored.
Default value: 0

- **score_per_conn=D** - a ratio between general score for the certain IP address and the number of connections from this IP address. The filter is applied only when the parameter value is not 0, and at least one connection from this IP was established. Default value: 100.0

If both parameters are specified, the **score_per_msg** parameter is checked first and the **score_per_conn** parameter is checked after it. If values of both parameters are set to 0, the filter is ignored.

Default values for general parameters:

- **min_msgs=0**
- **min_errors=0**
- **min_wrong_rcpts=0**
- **min_conn=100**
- **block_period=2h**
- **score=0**

Example:

```
ReputationIPFilter = errors_filter score=20, score_filter
```

The first filter sets a score equal to 20 to all messages in sessions established from IP addresses, which generate too many errors during the SMTP session. The second filter blocks all IP addresses, which have too large average scores in comparison to the number of connections established from them.

Example:

```
ReputationIPFilter = errors_filter errors_per_msg=0.05 errors_per_conn=1  
min_msgs=0 min_errors=10 min_wrong_rcpts=3 min_conn=50, score_filter  
score_per_msg=20 score_per_conn=30 min_wrong_rcpts=3, anti_dha  
wrong_per_valid_rcpts=0.02 min_wrong_rcpts=20
```

In this example, **errors_filter** filter is triggered when one of the following conditions is true:

- ratio between the number of errors occurred during the SMTP session and the number of messages passed to `drweb-maild` equals to 0.05 (`errors_per_msg=0.05`);
- ratio between the number of errors occurred during the SMTP session and the number of connections from this IP address equals to 1 (`errors_per_conn=1`);
- number of errors registered on the stage of SMTP session is greater than 10 (`min_errors=10`);
- number of invalid recipients (declined after the `RCPT TO` command) transferred by the SMTP client equals to 3 (`min_wrong_rcpts=3`);
- minimum number of connections from the IP address equals to 50 (`min_conn=50`).

score_filter filter is triggered if:

- ratio between the general score for the certain IP address and the number of messages passed to **MailD core** equals to 20 (`score_per_msg=20`);
- ratio between the general score for the certain IP address and the number of connections from this IP address equals to 30 (`score_per_conn=30`);
- number of invalid recipients (declined after the `RCPT TO` command) transferred by the SMTP client equals to 3 (`min_wrong_rcpts=3`).

anti_dha filter is triggered if:

- ratio between the number of invalid message recipients (declined after the `RCPT TO` command) and the number of valid recipients equals to 0.02 (`wrong_per_valid_rcpts=0.02`);
- number of invalid recipients (declined after the `RCPT TO` command) transferred by the SMTP client



equals to 20 (`min_wrong_rcpts=20`).

Please note that when an IP address is checked by `SessionRestriction`, on the connection stage only those scores are considered that were counted for this IP address during the previous session and the current stage. So, `min_conn` counter is always activated before the others. If the IP address passed `SessionRestriction` restriction but on next session stages its scores exceed the thresholds specified in the **Reputation IP Filter** settings, then the IP address is not blocked until the next session for this IP address is established.

Simultaneous Use of Several Receiver/Sender Components

You can connect several **Receiver** and/or **Sender** components to `drweb-maild` simultaneously.

This feature can be used for the following purposes:

- to enable concurrent interaction with several MTAs or **SMTP/LMTP proxy**;
- to enable differentiation of settings for each **Receiver/Sender** pair (which allows using different settings for monitored interfaces);
- to enable redirection of messages from one MTA to another (i.e. for routing).



Note that functions of **Sender** and **Receiver** components can be performed by different executable modules (for example, not only `drweb-receiver` module can be used as **Receiver**, but also `drweb-milter` or `drweb-cgp-receiver` depending on the used method of integration with MTA).

A complete list of modules and their roles (**Sender**, **Receiver**) in **Dr.Web MailD** are provided in the [Used Modules](#) section.

In the present section, it is assumed that `drweb-sender` module performs the role of **Sender** and `drweb-receiver` module - of the **Receiver**.

To enable simultaneous usage of several components:

1. Assign a unique identifier to each **Sender** and **Receiver** component (each element within the **Sender** or **Receiver** group must have a unique identifier, but ID of a **Receiver** must match an ID of a **Sender**).
2. Define the configuration source for each component.
3. Assign the ID of the **Receiver** that got the message as a tag.
4. After a message is processed, `drweb-maild` searches for an available **Sender** with the same ID as the **Receiver**'s. If not found, the message is dispatched to the default **Sender** (the only Sender with no unique ID specified) which must be always available.
5. The list of available **Sender** components is generated on the startup and is refreshed upon receipt of `SIGHUP` signal.
6. Routing of messages generated by `drweb-notifier` is managed by the `MsgIdMap` parameter from the `[Notifier]` [section](#) of **Dr.Web MailD** configuration file. This parameter allows defining to which **Sender** reports are to be sent in response to messages from certain **Receivers**.

Unique identifier for **Receiver/Sender** is set via `--unique-id` [command line parameter](#). When components are started with this parameter, they create in `%var_dir/messages/{in|out}` directory a number of subdirectories for their message queues, and in `%var_dir/ipc/` directory a special UNIX socket is created for **Sender**.

When the second copy of the component (for example, `drweb-receiver`) is started, an additional adjustment is required: that is, to specify the way this second copy receives configuration.

A component can receive configuration in the following ways:

- create a new copy of the `*.conf` file;



- modify the existing copy of the *.conf file (it is easier, but less flexible).

To modify an existing *.conf file:

- create a new *.amc file for **Dr.Web Agent** and add information on a new copy of the component. The file name is arbitrary.

Example:

```
Application "MAILD"
id 40
ConfFile "/etc/drweb/maild_smtp.conf"
Components
drweb-sender2 General, Logging, Sender2
  drweb-receiver2 General, Logging, /Maild/
    ProtectedNetworks, /Maild/
    ProtectedDomains, \
    /Maild/IncludeSubdomains, SASL,
    Receiver2
drweb-sender3 General, Logging, Sender3
  drweb-receiver3 General, Logging, /Maild/
    ProtectedNetworks, /Maild/
    ProtectedDomains, \
    /Maild/IncludeSubdomains, SASL,
    Receiver3
drweb-sender4 General, Logging, Sender4
  drweb-receiver4 General, Logging, /Maild/
    ProtectedNetworks, /Maild/
    ProtectedDomains, \
    /Maild/IncludeSubdomains, SASL,
    Receiver4
drweb-sender5 General, Logging, Sender5
  drweb-receiver5 General, Logging, /Maild/
    ProtectedNetworks, /Maild/
    ProtectedDomains, \
    /Maild/IncludeSubdomains, SASL,
    Receiver5
```

In the example, **drweb-receiver*** and **drweb-sender*** are the new names of the components used for interaction with **Dr.Web Agent**; **Receiver*** and **Sender*** are the new names of the corresponding section in the [configuration file](#).

Other parameters must be copied from the section with the original component settings. Detailed description of *.amc files syntax can be found in [Dr.Web Agent description](#).

- copy the main section with component settings to the *.conf file, rename this section (specify the name that was set on the previous step) and adjust all other settings in the new section for the second component;
- start or restart **Dr.Web Agent** to enable it to read the new configuration;
- start the new component with the following [command line parameters](#): --unique-id, --component, --section.

Example:

```
drweb-receiver --unique-id id2 --component drweb-receiver2 --section Receiver2
drweb-sender --unique-id id2 --component drweb-sender2 --section Sender2
```

To create a new copy of *.conf file (requires more effort but allows adjustment of different section settings):

- create a copy of original *.conf file and adjust parameters (it is not required to rename the sections!);



- create a new `*.amc` file that contains only information on the new component. You must also specify the path to the new `*.conf` configuration file created on a previous step;
- start or restart **Dr.Web Agent** to enable it to read the new configuration;
- start the new component with the following [command line parameters](#): `--unique-id`, `--component`.

Example:

```
drweb-receiver --unique-id id2 --component drweb-receiver2
drweb-sender --unique-id id2 --component drweb-sender2
```

Dr.Web Monitor can be configured to use new components for both ways of initialization. To do that, add the corresponding lines (about new components startup) to the `*.mmc` file of **Dr.Web MailD**.

Detailed description of `*.mmc` file syntax is provided above, in the [Dr.Web Monitor description](#).

Optimizing Operation and Use of System Resources

Thread pool control

As **Dr.Web MailD** [modules](#) use multithread model when receiving, processing or delivering a message, each of the modules create a certain number of processing threads. The more mail traffic is to be processed and the more messages are to be checked (for example, when **Drweb plug-in** scans a large number of messages in the paranoid mode or a large number of [Rules](#) are checked for matching), the more threads each component create. All created threads are organized in pools which behavior is controlled by [parameters of the PoolOptions](#) type. These parameters also set a number of threads in each pool (both minimum `t_min` and maximum `t_max` values). By default, the `auto` value is set for all thread pools created by all of the modules.



The `auto` value specified for a pool, sets the following values for `t_min` and `t_max`:

- For **drweb-receiver** and **drweb-sender** modules: `t_min=2`, `t_max=500`;
- For other modules (**drweb-maild**, **drweb-milter**, **drweb-notifier** and others): `t_min=2`, `t_max=1000`.

Restriction on the thread number set for **drweb-receiver** not only prevents the module from creating more active threads than specified but also influences the module behaviour during SMTP sessions. If the number of connections from clients exceeds the allowed limit of threads in a pool, the module creates maximum number of threads and all other connections, for which a processing thread cannot be created, are queued. Once an active thread becomes free, it starts processing a queued connection. As processing of connections is asynchronous, the same thread can process several connections simultaneously. Queue length of client connections is always restricted to the maximum allowed number of threads in the **drweb-receiver** pool. Thus, **drweb-receiver** can simultaneously handle no more than $2 * t_{max}$ connections, at that, some of them can be queued. Once the queue is full, **drweb-receiver** stops receiving new connections and responds to clients with the following error:

```
Server error: 421 3.8 Too many concurrent SMTP connections; please try
again later
```

On heavy load, some new connections are not discarded, as active threads start processing queued connections as soon as they become free and thus, other new connections can be queued. Nevertheless, it is recommended to increase the maximum limit (`t_max`) of threads in a pool of **drweb-receiver** module to avoid failure to process new connections. For other **Dr.Web MailD** modules, increase in number of threads does not influence module operation.

To control the components (number of active threads in pools and queue length), it is recommended to periodically send a `SIGUSR1` signal to [all processes](#) of **Dr.Web MailD**.



Dr.Web Monitor and **Dr.Web Agent**, which control operation of **Dr.Web MailD** components, do not process `SIGUSR1` signal in the current version. Thus, `SIGUSR1` signal, if sent to the components, causes them to terminate their operation!

When **Dr.Web MailD** components receive `SIGUSR1` signal, they reset statistics on thread pools. Statistics can be saved either to separate text files or to the log (on `Debug` level). Location of files with statistics is controlled by the `BaseDir` parameter from the [General] [section](#). For details on statistics format, refer to [Internal Statistics](#).

Statistics on thread pools of `drweb-sender` and `drweb-receiver` modules is saved to `sender_thr.txt` and `receiver_thr.txt` files respectively. Statistics contain data on the current size of the pool, number of active threads and queued connections. It is recommended to increase the maximum allowed number of pools (`t_max`) when the number of queued connections (`pending`) is approaching to the number of active threads (`active`).



Actual number of threads (for example, if counted with `ps aHx` command) is always greater than the number specified in the pool settings. That is because pool settings define only the number of processing threads, but during operation helper threads are also created.

It is required to increase the maximum limit of threads with caution. Before you change the setting, estimate:

- Amount of used memory;
- Number of files and sockets to be open (that is, file descriptors);
- CPU power.

The greater the `t_min` value is specified (determines the number of threads in a pool), the more time is required by **Dr.Web MailD** components to start and establish connections. For example, if **Drweb anti-virus plug-in** is used, threads from the plug-in thread pool establish connection with **Dr.Web Daemon** on the plug-in startup; therefore, time period which **MailD core** needs to start increases. If the minimum number of threads in a component pool is too large, time required to start the component can exceed the `startTimeout` parameter value specified in the **Dr.Web Monitor** settings for the component startup. In this case, **Dr.Web Monitor** abnormally terminates operation of both the component and the whole **Dr.Web MailD** software suite on startup.

Similarly, if too large number is specified as `t_max` value (maximum number of threads in a pool), errors can occur on termination of **Dr.Web MailD** suite when the period required for its components to shut down exceeds the timeout value. In this case, operation of the suite is terminated abnormally by **Dr.Web Monitor**.

It is not recommended to increase the maximum limit of threads as a reserve, because if that number is too large (about 1000 for `drweb-receiver` and `drweb-sender` modules and about 2000 for others), that may cause a delay in creation of new threads and lead to time-out errors while message processing. Such situation can cause processing errors and message loss. If so, decrease the number of threads. If it is impossible, do the following:

- 1) Increase the time-out value of the IPC subsystem (controlled by the `IpcTimeout` parameter in the [General] [section](#)), for example, to 10 minutes;
- 2) Increase the maximum allowed time to wait a thread to close, which is used on **Dr.Web MailD** startup and shutdown (controlled by the `MaxTimeoutForThreadActivity` parameter in the [General] [section](#)), for example, to 3 minutes;
- 3) Increase the time-out value to wait **Dr.Web MailD** components to start or shut down in the `maild_<mta>.mmc` [control file](#) of **Dr.Web Monitor** (as larger number of threads requires more time to stop).

In this case, it is also strongly recommended to adjust parameters of the whole complex (see below).



Possible symptoms of system resources exhaustion

- 1) It may occur that the successive thread in a pool cannot be created. If so, the following error is logged in a log of the corresponding component:

```
ERROR <some description>: boost::thread_resource_error
```

In this case, decrease the number of active threads for the corresponding thread pool. When it is set automatically (`auto`), specify the thread number explicitly.

If the specified number is not sufficient and increase in number of threads causes an error, increase server performance, that is, install more RAM and increase number of cores available for **Dr.Web MailD**.

- 2) On heavy load, processing of messages cannot be performed. If so, **Dr.Web MailD** logs the following error:

```
Too many open files
```

The error occurs because of exhaustion of file descriptors available for **Dr.Web MailD** (including socket descriptors).

To solve the problem (on **Solaris** OS 10), before `drweb-receiver` startup define the `LD_PRELOAD_32` environment variable and assign the following value: `/usr/lib/extendedFILE.so.1` to it. You can do that:

- directly in the console, if `drweb-receiver` is started not with the starting script, but from the console;
- by "wrapping" the startup of `drweb-receiver` into the script wrapper which sets the required value to this environment variable;
- by changing the start script for `drweb-monitor` (`/etc/init.d/drweb-monitor`) and adding the corresponding strings that change the system environment variable.

Note that in the last case the environment variable will be defined not only for `drweb-receiver`, but for all **Dr.Web** processes run by **Dr.Web Monitor**.

If that does not fix the problem, leave the made changes and do the following:

- increase the `ulimit -n` values;
- add (or adjust, if already exist) the following lines in the `/etc/system` file:

```
set rlim_fd_max = 65335
set rlim_fd_cur = 65335
```

If this error occurs on other OS (**FreeBSD** or **Linux**), increase the limit by the number of file descriptors for the process/user and increase the `ulimit -n` values.

General recommendations on how to enhance performance

To enhance performance on heavy load, it is recommended to:

- Use **asynchronous mode** of message processing, that is, assign the **plug-ins** to the **queues** as follows:

```
BeforeQueueFilters = headersfilter, vaderetro
AfterQueueFilters = drweb, modifier
```



Plug-ins assigned to **BeforeQueueFilters** queue can interact with **drweb-receiver** module synchronically and process messages before they are moved to the database. But if most of the messages have a large attachment (or large number of attachments), their processing by the plug-ins takes considerable time. In this case, it is not recommended to assign the plug-ins to **BeforeQueueFilters** queue as it can slow down interaction with external MTA when transmitting messages.

Moreover, in this case, a problem can occur while checking messages due to incorrect timeout value (too small) specified in the **IpcTimeout** parameter. The problem can cause message loss (it will not be deleted and the sender will not be notified on that).

If **Dr.Web MailD** is integrated with an MTA, it is recommended to use synchronous mode (plug-ins must be assigned to **BeforeQueueFilters** list). Otherwise, if it operates as a SMTP/LMTP proxy, it is recommended to use asynchronous mode (plug-ins must be assigned to **AfterQueueFilters** list). Both parameters are presented in the [Filters] section.

- Increase timeout values:
 - IPC subsystems (managed by the **IpcTimeout** parameter in the [General] section);
 - Maximum allowed time to wait for a thread to close; used on restart and shutdown of **Dr.Web MailD** operation (managed by the **MaxTimeoutForThreadActivity** parameter in the [General] section);
 - Time to wait for **Dr.Web MailD** components to start or shutdown in the `maild_<mta>.mmc` control file of **Dr.Web Monitor**.
- Increase the `ulimit -n` values.
- Estimate load on the thread pools by gathering and analyzing statistics (see above). If required, adjust the limits for the corresponding thread pools.
- Mount `%var_dir/msgs` and `%var_dir/infected` directories to the **tmpfs** file system (with the following command `mount -t tmpfs tmpfs <directory>`, where `<directory>` – mounted directory).



Mount directories to the **tmpfs** file system with caution. Note the following information:

- The system must have sufficient RAM memory;
- If power loss occurs, both external queues and content of **Quarantine** will be lost.

- For all parameters that contain Lookup, use `Lookup` to files (`file:`, `rfile:`), regular expressions (`regex:`) or lists (as contacting external DBMS and LDAP while processing each message considerably reduces processing speed and success of the processing depends on stability of connection to DBMS and or LDAP server).
- Set the value of the **MoveAll** parameter in the [Quarantine] section to `No` (especially if `%var_dir/msgs` and `%var_dir/infected` are not mounted to **tmpfs**).
- Set the value of the **SyncMode** parameter in the [MailBase] section to `No`.
- Increase the memory available for the internal DB. For that purpose, increase the **MaxPoolSize** parameter value in the [MailBase] section, which reduces the number of disk access requests.
- Disable use of statistics and reports (by setting the following parameter values **Detail=off** and **Send=no** in the [Stat] section and [Reports] section respectively).
- Configure logging to files instead of **syslog**.
- Specify protected networks and domains as a list in the **ProtectedNetworks** and **ProtectedDomains** parameter values in the [Maild] section (see the note on `Lookup` usage mentioned above).
- If no message processing Rule contains client-ip parameter in the conditional part, set the **GetIpFromReceivedHeader** parameter value in the [Maild] section to `No`.



- Set the following parameter values: **SkipDSNOnBlock** = Yes (in the [Maild] [section](#)), **SendSDN** = No (in the [Sender] [section](#)), and try to avoid **notify** and **redirect** [optional actions](#) in [plug-in](#) settings.
- Disable **Quarantine** (remove **quarantine** action from [plug-in](#) settings, if the action is specified).
- Limit the maximum size of messages checked by plug-ins (by setting required values to the **MaxSizeBeforeQueueFilters** and **MaxSizeAfterQueueFilters** parameters in the [Filters] [section](#)).
- Values of the **StalledProcessingInterval** parameters in the [Sender] [section](#) and [Receiver] [section](#) must not be less than the defaults (10m).

If during **Dr.Web MailD** operation delay in sending messages occurs and number of queued connections increases for the **drweb-sender** thread pool, do one of the following (depending on the delivery method set in the **Method** parameter in the [Sender] [section](#)):

- For SMTP delivery method:
 - Decrease the timeout value set in the **OtherCmdsTimeout** parameter in the [Sender] [section](#).
 - If value of the **Router** parameter in the [Sender] [section](#) is specified, avoid using **Lookup** that contact external DBMS and LDAP (see the note on **Lookup** usage, mentioned above).
 - Check operation of MTAs that receive messages from **Dr.Web MailD** – check time required to receive MTA response on attempt of **drweb-sender** module to establish connection as well as where an error occurs while message delivery.
- For Pipe delivery method:
 - Check operation of the local MTA that receives messages from **Dr.Web MailD**. Its daemon that is responsible for local message delivery must be configured correctly.

If during **Dr.Web MailD** operation number of messages in delivery queue increases (located in the `%var_dir/msgs/out` directory), it is recommended to send **SIGUSR2** [signal](#) to **drweb-sender** module when it is not peak usage time.

Moreover, you can implement [cluster solution](#) using internal proxying of requests from **Sender** and **Receiver** to several **MailD core** instances.

Recommendations on configuring Dr.Web MailD if the processed traffic primarily consists of large messages:

1. It is recommended to avoid using **Dr.Web Modifier** as well as any filtering based on content analysis (that is, avoid setting values to the **RejectPartCondition**, **AcceptPartCondition** and **MissingHeader** parameters of **Dr.Web HeadersFilter** plug-in as well as to the **RegexForChecked** parameter to **Drweb** plug-in and etc.). It is recommended because search within MIME objects can significantly reduce message processing. **Vaderetro**, **Dr.Web Modifier** and **Drweb** plug-ins store both message body and headers in RAM memory while processing, so it is recommended to assign the plug-ins to **AfterQueueFilters** queue (use [asynchronous](#) mode).
2. Increase IPC timeout (the **IpcTimeout** parameter in the [General] section) to 5 minutes, if **Drweb** plug-in is used.
3. Increase file scanning timeout in [Dr.Web Daemon settings](#) as well as the value of the **Timeout** parameter in **Drweb** plug-in settings (maximum 10 minutes).
4. Mount directory with messages and **Quarantine** directory (`%var_dir/msgs` and `%var_dir/infected`) into the **tmpfs** file system, but only if sufficient RAM memory is available in case of large number of messages.
5. Set the maximum size of a message body saved to the internal DB to 1 KB (set **MaxBodySizeInDB** = 1k in the [MailBase] section).
6. Stop restricting message size and amount of disk space available for **Quarantine** (set 0 as a value



of the `MaxSize` and `MaxNumber` parameters in the `[Quarantine]` section). Note that if DBI is used, **Quarantine** with large messages will be saved to DBMS, that will cause additional load on server. It does not influence **Dr.Web MailD** operation directly, but can affect the average load on the server.

7. Restrict time to store messages in **Quarantine** if no special condition is required (controlled with the `storedTime` parameter in the `[Quarantine]` section). Otherwise, they will consume disk space.
8. Control disk space consumed by `%var_dir/msgsg/out` and `%var_dir/msgsg/out/failed` directories, which allows to detect problems with message delivery and keep free space on the disk.

For stable **Dr.Web MailD** operation, it is required to have more RAM memory than total amount of messages processed per second and multiplied by the average time required for processing, in seconds. At that, limitation on the number of threads in a pool, specified in component pool settings, restricts the total number of messages per second. The average time required for processing one message in seconds depends only on server capacity (with constant settings for [plug-ins](#), [processing Rules](#) etc.). Thus, the average time must be estimated with the use of **Dr.Web MailD** logs for the current architecture. To provide stable operation, pools of `drweb-receiver` (`drweb-milter`), `drweb-maild` and `drweb-sender` modules must have the same number of threads. However, if `drweb-sender` uses routing configured by the `Router` [parameter](#) value, this module must have more threads in its pool than other modules.

If a time period of 5 seconds, specified in `IpcTimeout`, is not enough for message processing, balance the load and decrease the number of threads in pools (see above).

Using Internal Proxy

Internal proxy included in **Dr.Web MailD** allows to achieve several goals in management of computing resources:

1. **Dr.Web for UNIX mail servers** efficiency may improve greatly, when **Receiver** and **Sender** components are working separately from **MailD core** component (`drweb-maild` module), so that mail processing and mail checking operations are performed on different hosts.
2. Computing resources in a network can be managed flexibly, using load balancing scheme $N:M$, where N is the number of hosts processing mail traffic, and M is the number of hosts checking mail for viruses and spam.



Please note that `drweb-maild` module does not support cluster implementation and different module instances cannot share internal data (statistics, **Quarantine**, database settings, etc.).

As a result, each `drweb-maild` module of the M group will have its own statistics, **Quarantine**, and configuration.

Proxy consists of the following components: **Proxy-client** (`drweb-proxy-client` module) and **Proxy-server** (`drweb-proxy-server` module).

- **Proxy client** works on a host, where **Receiver** and **Sender** components are operating. It is started instead of **MailD core** and plays its role in interaction with other components.
- **Proxy server** works on a host, where the **MailD core** component is operating, and plays the role of **Receiver** and **Sender** components.

Both **Proxy client** and **Proxy server** components interact with each other, enabling transfer of original mail messages and their modifications to other components of **Dr.Web for UNIX mail servers** on different hosts for further processing.



Note that role of **Sender** and **Receiver** components can be played by different executable modules (for example, functions of **Receiver** can be performed not only by `drweb-receiver`, but also `drweb-milter`, `drweb-cgp-receiver` depending on the method of integration with MTA selected during **Dr.Web MailD** installation and adjustment).

Full list of the modules and their roles (**Sender**, **Receiver**) in terms of **Dr.Web MailD** is provided in the [Used Modules](#) section.

Dr.Web Notifier, **Dr.Web Monitor** and **Dr.Web Agent** components are working on each host.

General operation scheme with the use of a proxy is as follows (N=2, M=3):

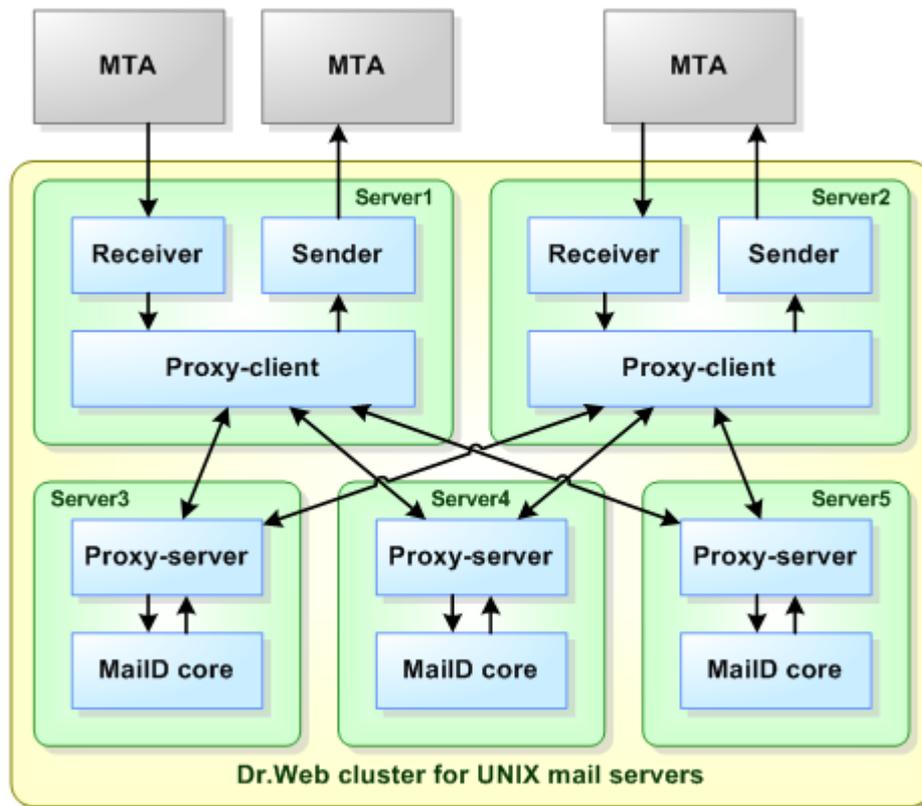


Figure 18. Diagram of Dr.Web MailD operation using a proxy

As it appears from the scheme, both **Proxy client** and **Proxy server** can interact with an arbitrary number of complementary components residing on different hosts. This is implemented using a special balance system.

A certain weight is assigned to each socket address specified in a value of the **ProxyServersAddresses** or **ProxyClientsAddresses** parameters (from the `[ProxyClient]` and `[ProxyServer]` [sections](#) respectively). So, addresses are specified in the following format:

```
ADDRESS1 [WEIGHT1], ADDRESS2 [WEIGHT2] ...
```

where **ADDRESS** has a basic address type, and **WEIGHT** is an optional numeric value from a range of 0 to 100, defining a weight of this address. This **WEIGHT** determines a relative work load on a certain host in the network. The greater the value is specified, the greater the load is on a certain server.

The **ProxyServersAddresses** parameter in the `[ProxyClient]` section on **Server1**, **Server2** hosts specifies addresses of **Server3**, **Server4**, **Server5** (see the scheme above), which are used by **Proxy-server** components to receive requests.



The `ProxyClientsAddresses` parameter in the `[ProxyServer]` section on **Server3**, **Server4**, **Server5** hosts specifies addresses of **Server1**, **Server2** (see the scheme above), which are used by **Proxy-client** components to receive requests.

Example:

```
ProxyServersAddresses = inet:8066@10.3.0.73 10, inet:8066@10.3.0.72 5
```

In this case, `10.3.0.73` host will receive twice as much mail messages as `10.3.0.72` host does (5 and 10 messages out of 15 respectively).

If the `WEIGHT` is not specified, it is considered to be equal to 1 by default. If several addresses have the same `WEIGHT`, they are considered equivalent and receive the same amount of requests.

If the `WEIGHT` is set to 0, such addresses are considered backup-addresses. Requests are sent to them only if no other available addresses with `WEIGHT`s equal to or greater than 1 are left.

General address selection algorithm looks as follows:

- 1) Randomly selects an address according to weights (than more weight the more probability to selection).
- 2) Attempts to send a message to the selected address.
- 3) If the message was sent successfully, the algorithm ends. Otherwise, one of the following is selected:
 - another address with the same weight (if exists),
 - address with less weight (weight must be not less than 1),and goes to step 2. If there are no more not-tried addresses (with weight not less than 1), the algorithm goes to step 4.
- 4) Attempts to send the message to backup addresses (in accordance with the order they are specified). If no address from the backup list is accessible, an error is returned.

`WEIGHT` values should be selected and assigned according to the resources available on each server, that is, assign greater `WEIGHT` values to hosts with more resources.

When messages are passed for check to the `drweb-maild` module and processed by plug-ins from the `BeforeQueue` `queue`, all processed mail is returned to the client that sent it for processing. If messages are processed by plug-ins assigned to `AfterQueue`, the address of the client that receives already processed mail will be selected according to the weights of client addresses in `ProxyClientsAddresses`. Duplicated messages (see the [Rules](#) description) and messages generated by `drweb-maild` (reports and notifications) will be also sent to the client selected from the `ProxyClientsAddresses` list - regardless of the queue in which the plug-ins reside. For messages sent to the client selected from the `ProxyClientsAddresses` list, settings specified in [Rules](#) (if there are any) will be applied (for example, value of the `SenderAddress` parameter).



Note that when a proxy interacts with **Qmail**, **Courier** MTAs (and other MTA using `Milter` protocol), it is better not to assign plug-ins to the `AfterQueue`. Currently a proxy does not support callback connections to `drweb-milter`. So, if the response from `drweb-maild` is not returned immediately (for example, when the plug-in is in the `AfterQueueFilters` and operation mode is `asynchronous`), `drweb-milter` finishes SMTP session only after expiration of the `ProcessingTimeout` period.

Optimal connection scheme for $M=N=1$ is described below (the scheme is preferable to avoid various errors in configuration):



1. Set up and adjust **Dr.Web for UNIX mail servers** on **Server1** (that is, on the host that is used for mail traffic processing and where **Proxy client** component resides). Check validity of the configuration with the following command:

```
/etc/init.d/drweb-monitor check (for Linux and Solaris)  
/usr/local/etc/rc.d/00.drweb-monitor.sh check (for FreeBSD)
```

2. Run **Dr.Web for UNIX mail servers** on **Server1** and check if the mail is processed correctly.
3. Setup **Dr.Web for UNIX mail servers** on **Server2** (that is, on the host used for mail message check and where **Proxy-server** component resides). During setup you may skip configuration of **Receiver** and **Sender** components - they are not necessary on this host.
4. Adjust configuration parameters of **Server2** similarly to **Server1**.
5. Disable startup of **Receiver** and **Sender** components by commenting out the corresponding lines in the `mmc file` (from the `%etc_dir/monitor` directory) on the **Server2** and enable startup of **Proxy-server** (`drweb-proxy-server`).
6. Specify the IP address of **Server1** as a value of the `ProxyClientsAddresses` parameter from the `[ProxyServer]` [section](#) in the **Dr.Web MailD** configuration file on the **Server2**. This address must be the same that is specified as a value of the `Address` parameter from the `[ProxyClient]` [section](#). Mail will be sent to it.
7. Check validity of configuration on the **Server2** host with the following command:

```
/etc/init.d/drweb-monitor check (for Linux and Solaris)  
/usr/local/etc/rc.d/00.drweb-monitor.sh check (for FreeBSD)
```

If the configuration is correct, you can start **Dr.Web MailD** on the **Server2**.

8. Specify IP address of the **Server2** as a value of the `ProxyServersAddresses` parameter from the `[ProxyClient]` [section](#) in the **Dr.Web MailD** configuration file on the **Server1**. This address must be the same that is specified as a value of the `Address` parameter from the `[ProxyServer]` [section](#). Requests for message checks will be sent to in.
9. Disable startup of **MailD core** (`drweb-maild`) component by commenting out the corresponding line in the `mmc file` from the `%etc_dir/monitor` directory on the **Server1** and enable startup of **Proxy client** (`drweb-proxy-client`).

Note that at attempt to start simultaneously on the same host **Proxy-client** and **MailD core** components **Dr.Web Monitor** will finish its operation, and no components will be initialized. Information about this error will be logged.

10. Check validity of configuration on the **Server1** host with the following command:

```
/etc/init.d/drweb-monitor check (for Linux and Solaris)  
/usr/local/etc/rc.d/00.drweb-monitor.sh check (for FreeBSD)
```

If the configuration is correct, you can restart **Dr.Web MailD** - and all the mail will be transferred for check to the **Server2**.

11. You may also disable **Dr.Web Daemon** and **Dr.Web Updater** on the **Server1** (if there are no more **Dr.Web** products on the system). These modules are no longer necessary.

You can also apply this algorithm when `M` and/or `N` are greater than 1: just connect additional hosts as it is described above and edit values of the corresponding parameters (`ProxyClientsAddresses` from the `[ProxyServer]` [section](#) and `ProxyServersAddresses` from the `[ProxyClient]` [section](#)) in the configuration files on those hosts. Specify `WEIGHT` values for addresses with respect to their resources.



Note the following operation aspect of the proxy server if **Proxy client** is functioning on several hosts and **Receiver** on one of these hosts has the **ReturnReject = No** [setting](#).

In this case, if the proxy server rejects a message received from this client, the server generates DSN and sends it to a randomly selected proxy client. If it serves a subnetwork different to the one from where the message was transmitted, the DSN might be not delivered to the message sender due to the address being unavailable from this subnetwork.

Thus, it is recommended to avoid setting the **ReturnReject** parameter value to **No** when configuring proxying if the server has several clients serving different subnetworks and message non-delivery might occur.

Integration with Cyrus SASL

To enable SASL authentication via **Cyrus SASL** (`saslauthd`) service, do the following:

1. Configure and start **Cyrus SASL** (`saslauthd`) service.
2. Configure **Dr.Web MailD** to use **Cyrus SASL** (the settings are specified in the [\[SASL\] section](#) and the [\[Cyrus-SASL\] section](#)):

- o Enable SASL authentication and use of `cyrus` driver:

```
Use = yes
Driver = cyrus
```

- o Specify the path to the `libsasl` library as the `Lib` parameter value
- o Specify the path to the configuration file that manages authentication via `saslauthd` service as the `Path` parameter value; for example: `/etc/sasl2/maild` (without the `.conf` extension). This file must be located in the directory where `saslauthd` searches for configuration.



The directory used by **Cyrus SASL** to find the configuration file depends on the **Cyrus SASL** version and OS distribution.

- **Cyrus SASL** version 2.x searches for the file in `/usr/lib/sasl2/` directory
- **Cyrus SASL** version 2.1.22 and newer also searches for the file in `/etc/sasl2/` directory

Cyrus SASL of any version starts file search from the `/usr/lib/sasl2/` directory. If the configuration file is found in this directory, the search stops.

3. Create the authentication configuration file (in the given example – `/etc/sasl2/maild.conf`) and specify required parameters. The file structure is as follows: `<parameter>: value` pairs specified one per line. If the parameter can have several values, they must be separated by white spaces. The following parameters are mandatory:

- o `pwcheck_method` – password authentication method. The name of a module, used for authentication, must be specified here. Allowed values are:

Value	Authentication source
<code>saslauthd</code>	<code>saslauthd</code> daemon
<code>auxprop</code>	An auxiliary module which retrieves external data storages (databases, LDAP) for authentication data retrieving

- o `mech_list` – List of authentication mechanisms to be used. Allowed values are `plain`, `login`, `cram-md5`, `digest-md5` and `ntlm`.

Note that for `saslauthd` only `plain` and `login` mechanisms can be used.

The `saslauthd` daemon can retrieve authentication data from the system file `/etc/shadow`, also it can use PAM and IMAP server data. For details on how to configure



`saslauthd` to use a necessary data source, refer to **Cyrus SASL** documentation.

- When required to use data stored in a database or LDAP, set the `pwcheck_method` parameter value to `authprop` and specify the data source as the `auxprop_plugin` parameter value. The following values are allowed:

Plug-in	Used data source
<code>sasldb</code>	sasldb database (Berkeley DB for Cyrus SASL)
<code>sql</code>	MySQL, PostgreSQL and SQLite relational DBMS
<code>ldapdb</code>	LDAP

If you set the value to `sasldb`, specify the path to the used database as the `sasldb_path` parameter value. If not specified, the default path `/etc/sasldb2` is used.

If you set the value to `sql`, configure the following parameters:

Parameter	Description
<code>sql_engine</code>	Defines the used DBMS. Allowed values: <ul style="list-style-type: none">• <code>mysql</code> - MySQL;• <code>pgsql</code> - PostgreSQL;• <code>sqlite</code> - SQLite.
<code>sql_hostnames</code>	Defines the address for DBMS connection (hostname or hostname:port). When several DBMS servers are used, specify several addresses, separated by commas. Note: For MySQL DBMS, specify the "localhost" value to connect via the UNIX socket, or specify the IP address 127.0.0.1 to connect via the TCP socket
<code>sql_user</code>	Defines the username for database connection
<code>sql_passwd</code>	Defines the user password
<code>sql_database</code>	Defines the database name
<code>sql_select</code>	Defines the SELECT SQL statement used for retrieving user password as plain text. Important note: Do not enclose the SQL statement in quotes. To specify a macro (see below), use a single quotation character ('). For SQL statements, the following macros can be used (they will be replaced with the corresponding data received from the client): <ul style="list-style-type: none">• <code>%u</code> - Username.• <code>%r</code> - Realm (domain) to which the user belongs. It can be either KERBEROS realm, or FQDN of the host where SASL application is launched, or email domain (that is, part of an email address following the at sign "@")

If you set the value to `ldapdb`, configure the following parameters of LDAP usage:

Parameter	Description
<code>ldapdb_uri</code>	LDAP URI to be used. You can specify the following prefixes: <ul style="list-style-type: none">• <code>ldapi://</code> connecting via the UNIX socket• <code>ldap://</code> connecting via the TCP connection• <code>ldaps://</code> establish a secured TCP connection (TLS is used)
<code>ldapdb_id</code>	Login for authentication on the LDAP server (proxy authentication)
<code>ldapdb_pw</code>	Password (as plain text) for authentication on the LDAP server (proxy authentication)
<code>ldapdb_mech</code>	Authentication mechanism used by LDAP server



Parameter	Description
ldapdb_rc (optional)	Path to the file containing personal settings of local LDAP client (libldap). For example, in this file it is possible to define the client TLS certificate used for secured connection.
ldapdb_starttls (optional)	TLS usage policy. Two values are allowed – <code>try</code> and <code>demand</code> . When <code>try</code> value is specified, the LDAP client module tries to establish a secured connection and if this attempt fails, switches to unsecured mode. When <code>demand</code> value is specified and a secured connection cannot be established, connection to LDAP server is refused.

Examples:

1. The simplest configuration (`saslauthd` is used):

```
pwcheck_method: saslauthd
mech_list: plain login
```

2. Using `sasl` datasource:

```
pwcheck_method: authprop
auxprop_plugin: sasldb
mech_list: plain login cram-md5
sasldb_path: /etc/sasldb2
```

3. Using `PostgreSQL` database:

```
pwcheck_method: auxprop
auxprop_plugin: sql
mech_list: PLAIN LOGIN CRAM-MD5 DIGEST-MD5 NTLM
sql_engine: pgsqldb
sql_hostnames: 127.0.0.1, 192.0.2.1
sql_user: username
sql_passwd: secret
sql_database: dbname
sql_select: SELECT password FROM users WHERE user = '%u@%r'
```

Note that it is not always necessary to create an authentication configuration file for **Cyrus SASL**. If such a file is not created, default authentication settings and default authentication data source are used.

Notification Templates

Notification templates are represented as files which have the `.msg` extension. These files store an email message structure corresponding to `RFC 822` and can contain different headers. Template files are used by **Notifier** to generate service messages that can be one of the following types: MailD notifications, statistics reports, and DSN notifications.

In addition to plain text, you can also use macros in a template body by marking them with a `$` character. While a notification is generated from the template, they are replaced with real macros.

Notification processing

When **Dr.Web Mail** is processing a message, any plug-in can request a notification on any event (virus detection, processing error, message blocking). Notifications are created by **Notifier** (the `drweb-notifier` plug-in) which generates a message and then sends it via **Sender**. Moreover, **Sender** can request **Notifier** to generate a DSN message for the sender notifying on a delivery failure.

All notifications and reports, including DSN, are generated from the templates which **Notifier** searches for in the directory specified in the `TemplatesBaseDir` parameter value.



A notification can be one of the following three types:

- **MailD notifications sent on a certain message**

Notifier uses message processing Rules (for details on the Rules, refer to the [Message Processing Rules](#) section) to check whether it is required to generate a notification for each of the following participants:

- message sender;
- message recipients (individual notifications are sent to those recipients for whom different notification settings are specified);
- **Dr.Web MailD** administrator.

Name of the template file used for generation of notifications is formed by adding the `sender_`, `rcpts_`, and `admin_` prefixes respectively to the name of the event on which notification is generated and adding the `.msg` extension. Thus, a template file name corresponds to the following regular expression:

```
(admin|rcpts|sender)_(.*)\.msg.
```

For example, `sender_virus.msg` is a name of a template used for notification on virus detection. If a template with such a name is not found, an error occurs. Events on which MailD notifications can be generated and suffixes used to indicate a type of the event are listed in the table below.

Suffix	Notification reason
archive	Attached archive is not checked due to violation of archive check restrictions specified for the scanning Dr.Web Daemon
cured	Successful attempt to cure a threat detected in a mail attachment
error	Error occurred when checking the message
license	Failure to check the message due to violation of the license restrictions
malware	Detection of a malicious attachment
rule	Blocking a message by a Rule (either a MailD core message processing Rule or a rule used by or Dr.Web Modifier or Dr.Web HeadersFilter plug-ins)
skip	Skipping an attachment while scanning (e.g., password protected archive or encrypted file)
virus	Detection of a virus in an attachment



Note that

- Notifications of some types are sent only to certain recipients. For example, by default, notification on **skip** event is sent only to the message sender (`sender_skip.msg`). If required, notifications on this event can be dispatched to others if the template is copied and renamed (`rcpts_skip.msg` and `admin_skip.msg` for recipients and administrator respectively). However, it is also recommended to modify these templates so that they contain information suitable for a certain recipient. Templates available by default are listed in the table below.
- If several events occur during check of one message, **Notifier** sends a separate notification for each event to all recipients for whom that is allowed.
- It is possible to disable notifications of certain types to certain recipients depending on condition validation results. To do that, use the `notify` setting in [MailD core rules of message processing](#).

- **Periodic MailD notifications on total operation of the suite (reports for the administrator)**

Notifications of this type are sent by **Notifier** to the administrator. The notifications contain general statistics on the suite operation. The template is contained in the `report.msg` file.

- **Service notifications on a message delivery failure (DSN)**

Notifications of this type have a certain format and are generated to notify the message sender on a



delivery failure. DSN is always sent to the message sender and has an empty `FROM:` header. The used template is contained in the `dsn.msg` file.

In each case, the **Dr.Web for UNIX mail servers** component which requested notification sends information on the reason to the `drweb-notifier` plug-in. All templates except for DSN templates support the following two message types by default: HTML and plain text. Type of the notification message is selected according to the `html` parameter value specified in the [message processing Rules](#).

Note that MailD notifications and periodic administrator reports are sent from the address specified in the `FilterMail` parameter; at that, notifications, as well as DSN, are checked on matching the Rules.



Note that MailD notifications are dispatched to the message recipients, message senders, and the administrator by **Notifier** as a message sent from the address specified in the `FilterMail` parameter value. Service notifications DSN always have an empty `From:` field.

Name of a template file can be changed depending on certain criteria. For that purpose, the `NotificationNamesMap` parameter is used in message processing Rules. The parameter value defines how the name of a notification transmitted to **Notifier** is mapped to a new value from which a new template name is formed according to the above-mentioned pattern. It is reasonable to map a name to the one that **Notifier** can recognize, otherwise, the required file cannot be found. Such a situation is treated as an error and is processed according to the `ProcessingError` parameter value.



Note that the `NotificationNamesMap` parameter allows to configure selection of different user template files only for generating notifications of the second and third types; that is, only for periodic reports and DSN.

Example

```
[Rule:buh]
...
NotificationNamesMap = report r1, dsn d1
...
[Rules]
to:regex:*@buh.domain.org cont rule=buh
```

After this message processing Rule is applied, notifications of the second type (periodic reports) and DSN are generated from the `report_r1.msg` and `dsn_d1.msg` files respectively when mail messages are received from the `buh.domain.org` domain.



Please note that **Dr.Web MailD** is supplied with a standard `dsn.msg` template for DSN notifications and an additional `dsn_for_exchange.msg` template. The latter is a special DSN template used only if a target MTA is an **MS Exchange** mail server (required due to implementation features of **MS Exchange** that is not fully compliant with RFC 3464).

This special DSN **cannot be used with other MTAs**. If it is required to use `dsn_for_exchange.msg`, change the standard `dsn.msg` template to it with the following command:

```
cp dsn_for_exchange.msg dsn.msg
```

That allows you to avoid reconfiguration of **Notifier**.

If you may need to use the standard `dsn.msg` template in the future, save its copy before the change.

Instead of using the command described above, you can create a Rule that changes the `NotificationNamesMap` parameter value. That allows avoiding the change of the template files.

Templates Available by Default

By default, **Dr.Web for UNIX mail servers** is supplied with the following template files:



Template name	Description
Templates of reports sent to the administrator:	
ADMIN_ARCHIVE.msg	Template for a report generated on detection of archives which cannot be scanned due to excess of limits set for archives in main configuration file drweb32.ini
ADMIN_CURED.msg	Template for a report generated on cure of an infected message
ADMIN_ERROR.msg	Template for a report generated if Dr.Web Daemon or plug-in errors occur
ADMIN_LICENSE.msg	Template for a report generated when an email message cannot be checked due to license restrictions
ADMIN_MALWARE.msg	Template for a report generated on detection of malware in an email message
ADMIN_RULE.msg	Template for a report generated on rejection of a message due to a some rule
ADMIN_VIRUS.msg	Template for a report generated on detection of a virus in a message
Templates of notifications sent to the message recipients:	
RCPTS_MALWARE.msg	Template for a report generated on detection of malware in a message
RCPTS_VIRUS.msg	Template for a report generated on detection of a virus in a message
Templates of notifications sent to the message senders:	
SENDER_ARCHIVE.msg	Template for a report generated on detection of archives which cannot be scanned due to excess of limits set for archives in main configuration file drweb32.ini
SENDER_CURED.msg	Template for a report generated on cure of an infected message
SENDER_ERROR.msg	Template for a report generated on Dr.Web Daemon or plug-in errors
SENDER_MALWARE.msg	Template for a report generated on detection of malware in a message
SENDER_VIRUS.msg	Template for a report generated on detection of a virus in a message
SENDER_SKIP.msg	Template for a report generated on message scan failure. It can happen when password protected or broken archive or a file in a non-standard format is attached to a message, or when message scan is aborted due to timeout
Other templates:	
DSN.msg	Template for delivery status notification (DSN)
REPORT.msg	Template for regular Dr.Web Daemon reports

Used Macros

The following macros can be used in any template:

Macro	Description
\$LC*\$	Replaced with a string that have the specified number in a language files (* – decimal number of the string; for example,\$LC150\$). The used language file is determined by the \$LANG\$ macro. Conversion of the result text to the required encoding is configured by the values of the \$CHARSET\$ and \$CONTENT_TRANSFER_ENCODING\$ macros
\$POSTMASTER\$	Address to which all notifications are sent (value of the AdminMail parameter from the [Notifier] section is used)
\$FILTER_MAILS\$	Address used by Dr.Web MailD (value of the FilterMail parameter from the [Notifier] section is used)
\$HOSTNAME\$	Name of the host on which Dr.Web MailD is operating (value of the Hostname parameter from the [General] section is used). This macro cannot be used in loops; for details, refer to Control constructions
\$LANGS\$	Contains the list of languages on which notifications are generated (values



Macro	Description
	of the <code>NotifyLangs</code> parameter from the <code>[Notifier]</code> section are used). This macro can have a list as a value and can be used in loops; for details, refer to <code>Control constructions</code>
<code>\$LANG\$</code>	Name of the language on which this part of a notification is generated. The macro value defines interpretation of some other macros (for example, <code>\$CHARSET\$</code>)
<code>\$CHARSET\$</code>	Set of the current language characters. The character set is specified in the language file . Name of the currently used language is specified by the <code>\$LANG\$</code> macro
<code>\$CONTENT_TRANSFER_ENCODING\$</code>	Content-Transfer-Encoding for the current language. The value for a certain language is specified in the language file . Name of the current macro is specified by the <code>\$LANG\$</code> macro
<code>\$TYPE\$</code>	Notification content type (HTML or PLAIN). The content type is specified by the <code>html</code> parameter in the message processing Rules .
<code>\$FULLHEADERS\$</code>	Complete set of mail message headers.
<code>\$MSGID\$</code>	Internal message identifier in MTA which transmitted the message
<code>\$SUBJECT\$</code>	Message subject (empty if the subject is not specified). When inserted to the generated message, the macro value is converted, if required, according to <code>\$CHARSET\$</code> and <code>\$CONTENT_TRANSFER_ENCODING\$</code> macro values.
<code>\$DIRECT_SUBJECT\$</code>	Message subject (empty if the subject is not specified). Is not converted to another encoding or CTE when inserted.
<code>\$SENDER\$</code>	Address of the original message sender
<code>\$RCPTS\$</code>	List of all message recipient addresses. This macro can have a list as a value and can be used in loops; for details, refer to <code>Control constructions</code>
<code>\$SECURE_RCPTS\$</code>	This macro is similar to <code>\$RCPTS\$</code> if the message has only one recipient. Otherwise, the macro value is set to "Recipients of original message" <code><#@[]></code>
<code>\$LOG_REPORT\$</code>	Records from the Dr.Web MailD log file that contain information on processing of the message on which the notification is generated
<code>\$STOP_REASON\$</code>	Record from the Dr.Web MailD log file that contain the reason for sending this notification
<code>\$REPORT\$</code>	Plug-in report on analysis of the message on which the notification is generated
<code>\$MESSAGE_STATUS\$</code>	Status of the original message specified by POP3 and IMAP filters according to the processing results. This macro can have one of the following values: <code>reject</code> , <code>discard</code> , <code>tempfail</code> , and <code>error</code> .
<code>\$BLOCK_LIST\$</code>	List of strings describing reasons to block the message by a plug-in (more than one reason is possible). For example, if a threat is detected, Drweb plug-in returns the threat name. If the message is blocked due to another reason (for example, as a reaction to a <code>SkipObject</code> event), the plug-in returns a full value of the configuration string <code><parameter> = <value></code> , which caused the message block.
<code>\$SCAN_STAT\$</code>	Statistics on results of the plug-in message check



Macro	Description
\$ARCHIVE_RECORD\$	Name of the file in Quarantine
\$ORIGINAL_MESSAGE\$	Body of the original message for which notification is generated. Be careful when inserting the macro in a notification: if, for example, the original message body contained a virus, the generated notification could be blocked by another anti-virus system!
\$R_MAIL\$	List of addresses to which notifications are to be sent. The macro value is specified in the Mail parameter in the [Reports] section . This macro can have a list as a value and can be used in loops; for details, refer to Control constructions .
\$CO_CLIENT_IP\$	IP address of the client that transmitted the message (if known)
\$CO_CLIENT_PORT\$	Port number used by the client which transmitted the message (if Receiver provided this information)
\$CO_AUTH\$	The value is set to <i>yes</i> , if the client which sent the original message is successfully authorized (if Receiver provided this information)
\$CO_SERVER_UNIX_SOCKET\$	Name of the UNIX socket which was used by Receiver to get the original message (if Receiver provided this information)
\$CO_SERVER_IP\$	IP address of the listening socket which was used by Receiver to get the original message (if Receiver provided this information)
\$CO_SERVER_PORT\$	Port of the listening socket which was used by Receiver to get the original message (if Receiver provided this information)
\$CO_RS_ID\$	Identifier of Receiver which got the original message (if this Receiver instance was started with a non-empty identifier)
\$CO_SENDER_ADDRESS\$	Address specified in the SenderAddre parameter value in Rules for the message
\$Q_CONTROL_BY_EMAIL\$	The value is set to <i>yes</i> , if control messages for Quarantine management are allowed
\$TEMPLATES_DIR\$	Name of the directory with notification templates. The path is also used for search of all files specified in the template in the include directive ; for details, refer to Control constructions
\$Q_REMOVE_TIME\$	Time when the message was deleted from Quarantine (an empty string if the period of storing a message is not restricted)
\$PRODUCT\$	The following string: "MailD"
\$EXT_PRODUCT\$	The following string: "for Unix mail servers"

The following macros used in statistics reports

Macro	Description
\$R_PLUGIN\$	List of plug-ins for which statistics reports are generated. The value is specified in the Names parameter from the [Reports] section . This macro can have a list as a value and can be used in loops; for details, refer to Control constructions
\$R_PERIOD\$	Time period during which statistics reports are generated
\$RP_NAME\$	Plug-in for which the statistics report is generated
\$RP_BLOCKED_OBJECTS_WITH_NUM_AND_PERCENT\$	Calculated statistics on blocked objects for the plug-in specified in the \$RP_NAME\$ macro value
\$RP_SENDERS_ENVELOPE_WITH_NUM_AND_PERCENT\$	Statistics on senders of the blocked objects for the plug-in specified in the



Macro	Description
	<code>\$RP_NAME\$</code> macro value
<code>\$RP_CLIENT_IP_WITH_NUM_AND_PERCENTS\$</code>	Statistics on IP addresses blocked by the plug-in specified in the <code>\$RP_NAME\$</code> macro value
<code>\$RP_BLOCKED_OBJECTS_NUM\$</code>	Number of blocked objects included in the statistics. If the macro value is 0, objects are not output. If the value is -1, all blocked objects are output. In other cases, the specified number of objects is output. Depends on the <code>\$RP_NAME\$</code> macro value
<code>\$RP_SENDERS_ENVELOPE_NUM\$</code>	Number of senders included in the statistics on blocked objects. If the macro value is 0, senders of blocked objects are not output. If the value is set to -1, all senders of blocked objects are output. In other cases, the specified number of senders is output. Depends on the <code>\$RP_NAME\$</code> macro value.
<code>\$RP_CLIENT_IP_NUM\$</code>	The number of IP addresses included in the statistics on blocked objects. If the value is 0, no IP address is output. If the value is set to -1, all IP addresses are output. In other cases, the specified number of IP addresses is output.. Depends on the <code>\$RP_NAME\$</code> macro value.
<code>\$RP_TEMPFAIL_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>tempfail</code> action
<code>\$RP_PASS\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>pass</code> action
<code>\$RP_REJECT\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>reject</code> action
<code>\$RP_DISCARD\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>discard</code> action
<code>\$RP_TEMPFAIL\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>tempfail</code> action
<code>\$RP_REJECT_PLUS_TEMPFAIL\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied <code>reject</code> or <code>tempfail</code> action
<code>\$RP_QUARANTINE\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>quarantine</code> action
<code>\$RP_REDIRECT\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>redirect</code> action
<code>\$RP_NOTIFY\$</code>	Number of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>notify</code> action
<code>\$RP_PASS_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>pass</code> action
<code>\$RP_REJECT_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>reject</code> action
<code>\$RP_DISCARD_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>discard</code> action
<code>\$RP_REJECT_PLUS_TEMPFAIL_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>reject</code> or <code>tempfail</code> actions
<code>\$RP_QUARANTINE_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>quarantine</code> action
<code>\$RP_REDIRECT_SIZE\$</code>	Total size of message for which the plug-in specified in the <code>\$RP_NAME\$</code>



Macro	Description
	macro applied the <code>redirect</code> action
<code>\$RP_NOTIFY_SIZE\$</code>	Total size of messages for which the plug-in specified in the <code>\$RP_NAME\$</code> macro applied the <code>notify</code> action
<code>\$RP_BLOCK_PERC\$</code>	Total per cent of messages blocked by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_BLOCK_SIZE\$</code>	Total per cent of messages blocked by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_CHECK_TIME_SUM\$</code>	Total time required to check messages by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_CHECK_TIME_AVR\$</code>	Average time required to check messages by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_CHECKED_MSGS\$</code>	Total number of messages checked by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_CHECKED_SIZE\$</code>	Total size of all messages checked by the plug-in specified in the <code>\$RP_NAME\$</code> macro
<code>\$RP_AGENT_STAT_UUID\$</code>	UUID of the switch used by Dr.Web Agent for sending statistics to the Dr.Web statistics server or to the central protection server (the macro value is an empty string if this option is disabled)

Control Constructions

In the notification templates, the following control constructions can be used:

1. Custom Macro Declaration

If required, you can declare custom macros directly in the template by using the following construction:

```
<def NAME=DATA>
```

where `NAME` – macro name (without a `$` character), `DATA` – new macro value.

Name of the macro can consist of characters in the `[a-zA-Z_ -]` range. A new macro value can be quoted. The character that follows the back slash `"\"` is treated directly (that is, you can use this character to add, for example, a back angle bracket character).

Example

```
<def MY_MACROS="\<my macros\>">
```

In this example, the `MY_MACROS` macro that contains `'<my macros>'` is declared.

2. Inserting External File Content

You can add the external file content to the template by using the following directive

```
<include FILENAME>
```

where `FILENAME` – name and path to the file relative to the `$TEMPLATES_DIR$` [macro](#) value. File name and file path can be quoted. The character that follows the back slash `"\"` is treated directly (that is, you can use this character to add, for example, a back angle bracket character).

Example

```
<include "style.css">
```

In this example, text from the `style.css` file is inserted in the stead of the directive.



3. Conditional Operators

You can use conditional operators in the templates. The operators are defined as follows:

```
<if NAME [ ( '==' | '!=' ) DATA ] >
TEXT
</if>
```

where `NAME` – macro name, `DATA` – regular expression to check the macro value (regular **Perl** expressions are used), `TEXT` – text to be inserted into the message if the condition is true. The following two conditional operators are available:

- `==` – True if the macro value corresponds to the regular expression;
- `!=` – True if the macro value does not correspond to the regular expression.

If a part of the construction in square brackets is not specified (that is, the condition is `<if NAME>`), the construction is treated as `<if NAME != "" >`.

Use of nested conditional operators is allowed (for details, see below). If the operator includes the `def` directive to declare a new macro, this macro is defined only if the conditional part is true. At that, a new macro value is saved and becomes available after the conditional operator is checked.

Example

```
<def N="n123">
<if N>N is not empty!</if>
<if N == "n.*">N starts with n!</if>
<if N != n123>WRONG!</if>
```

In a notification generated from such a template, the following strings are inserted:

```
N is not empty!
N starts with n!
```

3. Loop

You can use loops in the templates. A loop is specified as follows:

```
<for NAME;LIST [ ( '==' | '!=' ) DATA ; [DELIM] ] >
TEXT
</for>,
```

where

- `NAME` – name of a macro that is used as a local variable in the loop body. Each time the loop iterates, another value retrieved from `LIST` is assigned to the variable.
- `LIST` – macro used as a list of values for the local variable in the loop. Macro value can be a list. In this case, any macro can be specified; at that, if its value cannot be a list, the value is separated with commas and transformed into the list.
- `DATA` – regular expression to check and select values from the list each time the loop iterates.
- `DELIM` – delimiter inserted in the notification body between text fragments generated each time the loop iterates.

If the values in square brackets are not specified, the construction is treated as `<for NAME;LIST == ".*" >`; that is, all values from `LIST` are used.

Otherwise, each value from `LIST` is compared with the expression specified in `DATA` and if the result is true, (`==` – corresponds, a `!=` – does not correspond), the next value is assigned to the `NAME` macro and the loop iterates. If `DELIM` is specified, the corresponding value selected as a delimiter is inserted into the notification.



Loop processing consists of the following steps:

- 1) For each selected NAME, the following text is generated

```
<def NAME="LIST_VAL">  
TEXT DELIM,
```

where LIST_VAL – next value selected from LIST, and TEXT – text specified in the loop body.

- 2) For this text a syntax analyzer is called and the outcome of its operation is placed immediately after the <\for> label.
- 3) This operation is performed for each selected value from LIST.
- 4) Loop construction is removed from the generated notification and syntax analyzer starts parsing the result text.

Example

```
<def RCPTS="root@localhost, test@mydoamin.com">  
<for RCPT;RCPTS==".*";", "><a href="mailto:$RCPT$">$RCPT$</a></for>
```

The result text is the following:

```
<a href="mailto:root@localhost">root@localhost</a>,  
<a href="mailto:test@mydoamin.com">test@mydoamin.com</a>
```

All key words `def`, `if` and `for` are case insensitive.

Template Example

The following example demonstrates a template used for generation of reports on plug-in operation. The template supports both HTML and PLAIN formats.



```
From: "DrWeb-$PRODUCT$" <$FILTER_MAIL$>
To: $R_MAIL$
Subject: Report from Dr.Web $PRODUCT$ per period of $R_PERIOD$
Content-Type: multipart/mixed;
  boundary="001-DrWeb-MailFilter-Notification"
MIME-Version: 1.0

<if TYPE==HTML>
<for LANG;LANGS>
--001-DrWeb-MailFilter-Notification
Content-Type: text/html; charset=$CHARSET$
Content-Transfer-Encoding: $CONTENT_TRANSFER_ENCODING$

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=$CHARSET$" />
<title>$LC510$</title>
<include "style.css">
</head>

<body>
  <div align="center">
    <table width="600" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td align="center" valign="top"><a name="top" id="top_$LANG$"></a>
          <include "head.templ">
          <table width="100%">
            <tr>
              <td align="right" ><for RP_NAME;R_PLUGINS=="*" "&nbsp;"><a
href="#"$RP_NAME$_$LANG$" class="ancherlinks">$LC543$ $RP_NAME$</a></for></
td>
            </tr>
          </table>
          <p class="titletext">$PRODUCT$: $LC542$ $R_PERIOD$</p>

          <for RP_NAME;R_PLUGINS>
          <table width="100%">
            <tbody>
              <tr>
                <th colspan="2" class="header"><a name="$RP_NAME$"
id="$RP_NAME$_$LANG$" ></a>$LC543$ $RP_NAME$</th>
              </tr>
              <tr>
                <td colspan="2" ><table width="100%" >
                  <tr>
                    <td colspan="2" ><table width="100%" >
                      <tr>
                        <td colspan="2" ><table width="300"
cellpadding="5" cellspacing="0" class="statistic" id="statistic2">
                          <tbody>
                            <tr>
                              <th colspan="2" class="statisticheader" >
                                <for RP_BLOCKED_OBJECTS_NUM=="-1">
                                  $LC544$:
                                </if><for RP_BLOCKED_OBJECTS_NUM!="-1">
                                  $LC545$ $RP_BLOCKED_OBJECTS_NUM$
                                </if>
                              </th>
                            </tr>
                          </tbody>
                        </tr>
                      </tr>
                    </td>$RP_BLOCKED_OBJECTS_WITH_NUM_AND_PERC
ENT$</td>
                    </tr>
                  </table>
                </td>
              </tr>
            </tbody>
          </table>
          </for>
        </td>
      </tr>
    </table>
  </div>
</body>
</html>
```



```

        </tr>
      </tbody>
    </table></td>
  </if></if>
  <if RP_SENDERS_ENVELOPE_WITH_NUM_AND_PERCENTS><if
RP_SENDERS_ENVELOPE_NUM!="0">
    <td valign="top"><table width="300"
cellpadding="5" cellspacing="0" class="statistic" id="statistic">
      <tbody>
        <tr>
          <th colspan="2" class="statisticheader" >
            <if RP_SENDERS_ENVELOPE_NUM=="-1">
              $LC547$:
            </if><if RP_SENDERS_ENVELOPE_NUM!="-1">
              $LC545$ $RP_SENDERS_ENVELOPE_NUM$
$LC548$:
            </if>
          </th>
        </tr>
        <tr>
          <td
class="regulartext">$RP_SENDERS_ENVELOPE_WITH_NUM_AND_PERCENTS$</td>
        </tr>
      </tbody>
    </table></td>
  </if></if>
  <if RP_CLIENT_IP_WITH_NUM_AND_PERCENTS><if
RP_CLIENT_IP_NUM!="0">
    <td valign="top"><table width="300"
cellpadding="5" cellspacing="0" class="statistic" id="statistic">
      <tbody>
        <tr>
          <th colspan="2" class="statisticheader" >
            <if RP_CLIENT_IP_NUM=="-1">
              $LC566$:
            </if><if RP_CLIENT_IP_NUM!="-1">
              $LC545$ $RP_CLIENT_IP_NUM$ $LC565$:
            </if>
          </th>
        </tr>
        <tr>
          <td
class="regulartext">$RP_CLIENT_IP_WITH_NUM_AND_PERCENTS$</td>
        </tr>
      </tbody>
    </table></td>
  </if></if>
</tr>
</table></td>
</tr>
<tr>
  <td class="body">$LC550$:</td>
  <td align="right" class="body">$RP_PASS$ ($RP_PASS_SIZE$)</
td>
</tr>
<tr>
  <td class="body">$LC551$:</td>
  <td align="right" class="body">$RP_REJECT$
($RP_REJECT_SIZE$)</td>
</tr>
<tr>
  <td class="body">$LC552$:</td>
  <td align="right" class="body">$RP_DISCARD$
($RP_DISCARD_SIZE$)</td>
</tr>

```



```

        <tr>
            <td class="body">$LC553$:</td>
            <td align="right" class="body">$RP_TEMPFAIL$
($RP_TEMPFAIL_SIZE$)</td>
        </tr>
        <tr>
            <td class="body">$LC554$:</td>
            <td align="right" class="body">$RP_QUARANTINE$
($RP_QUARANTINE_SIZE$)</td>
        </tr>
        <tr>
            <td class="body">$LC555$:</td>
            <td align="right" class="body">$RP_REDIRECT$
($RP_REDIRECT_SIZE$)</td>
        </tr>
        <tr>
            <td class="body">$LC556$:</td>
            <td align="right" class="body">$RP_NOTIFY$
($RP_NOTIFY_SIZE$)</td>
        </tr>
        <tr>
            <td class="subtitle">$LC557$:</td>
            <td align="right" class="subtitle">$RP_CHECKED_MSGS$
($RP_CHECKED_SIZE$)</td>
        </tr>
        <tr>
            <td class="subtitle">$LC571$:</td>
            <td align="right" class="subtitle">$RP_BLOCK_PERC$
($RP_BLOCK_SIZE$)</td>
        </tr>
        <tr>
            <td class="subtitle">$LC570$:</td>
            <td align="right" class="subtitle">$RP_CHECK_TIME_SUM$
(~$RP_CHECK_TIME_AVR$ $LC558$)</td>
        </tr>
    </tbody>
</table>
    <div>
        <if RP_NAME == "drweb" ><if RP_AGENT_STAT_UUID>
            <p align="right" class="regularText"> $LC567$ <a href="http://
stat.drweb.com/view/$RP_AGENT_STAT_UUID$">$LC568$</a>. </p>
        </if></if>
        <p> <a href="#top_$LANG$" class="ancherlinks">$LC561$</a> </p>
    </div>
</tr>
</table>
</div>
<br />
</body>
</html>
</for>
</if><if TYPE==PLAIN>
<for LANG;LANGS>
--001-DrWeb-MailFilter-Notification
Content-Type: text/plain; charset=$CHARSET$
Content-Transfer-Encoding: $CONTENT_TRANSFER_ENCODING$

    $LC542$ $R_PERIOD$

<for RP_NAME;R_PLUGINS==".*";">
    *** $LC543$ $RP_NAME$ ***
<if RP_BLOCKED_OBJECTS_WITH_NUM_AND_PERCENTS>
<if RP_BLOCKED_OBJECTS_NUM!="0">
<if RP_BLOCKED_OBJECTS_NUM=="-1">
$LC544$:

```



```
</if><if RP_BLOCKED_OBJECTS_NUM!="-1">
$LC545$ $RP_BLOCKED_OBJECTS_NUM$ $LC546$:
</if>
$RP_BLOCKED_OBJECTS_WITH_NUM_AND_PERCENTS$
</if>
</if>
<if RP_SENDERS_ENVELOPE_WITH_NUM_AND_PERCENTS>
<if RP_SENDERS_ENVELOPE_NUM!="0">
<if RP_SENDERS_ENVELOPE_NUM=="-1">
$LC547$:
</if><if RP_SENDERS_ENVELOPE_NUM!="-1">
$LC545$ $RP_SENDERS_ENVELOPE_NUM$ $LC548$:
</if>
$RP_SENDERS_ENVELOPE_WITH_NUM_AND_PERCENTS$
</if>
</if>
<if RP_CLIENT_IP_WITH_NUM_AND_PERCENTS>
<if RP_CLIENT_IP_NUM!="0">
<if RP_CLIENT_IP_NUM=="-1">
$LC566$:
</if><if RP_CLIENT_IP_NUM!="-1">
$LC545$ $RP_CLIENT_IP_NUM$ $LC565$:
</if>
$RP_CLIENT_IP_WITH_NUM_AND_PERCENTS$
</if>
</if>
    $LC549$:
    $LC550$:      $RP_PASS$ ($RP_PASS_SIZE$)
    $LC551$:      $RP_REJECT$ ($RP_REJECT_SIZE$)
    $LC552$:      $RP_DISCARD$ ($RP_DISCARD_SIZE$)
    $LC553$:      $RP_TEMPFAIL$ ($RP_TEMPFAIL_SIZE$)
    $LC554$:      $RP_QUARANTINE$ ($RP_QUARANTINE_SIZE$)
    $LC555$:      $RP_REDIRECT$ ($RP_REDIRECT_SIZE$)
    $LC556$:      $RP_NOTIFY$ ($RP_NOTIFY_SIZE$)

-----
    $LC557$:      $RP_CHECKED_MSGS$ ($RP_CHECKED_SIZE$)
    $LC571$:      $RP_BLOCK_PERC$ ($RP_BLOCK_SIZE$)
    $LC570$:      $RP_CHECK_TIME_SUM$ (~$RP_CHECK_TIME_AVR$ $LC558$)
<if RP_NAME == "drweb" ><if RP_AGENT_STAT_UUID>

    $LC567$ $LC568$:
        http://stat.drweb.com/view/$RP_AGENT_STAT_UUID$
</if></if>
</for>
</for>
</if>
--001-DrWeb-MailFilter-Notification--
```

Language Files

Language files are used as a source of text data (string resource) for generation of notifications, output of text messages or insertion of text fragments into processed email messages. Language files are used mainly by **Notifier**, but they can also be used by the plug-ins (for example, **Drweb plug-in** uses text from a language file to replace a malicious attachment with the text). Path to the directory where the language files are located must be the same for **Notifier** and the plug-ins. Path to this directory is specified in value of the **LngBaseDir** parameter (see above).

Pattern of a language file name is as follows

[<plug-in>_]<language>.lng, where:

- <plug-in> – name of the plug-in that uses this file as a string source (drweb, modifier and others)



- `<language>` – name of the language used in the file.

Language files used by **Notifier** do not have the `<plug-in>_` prefix in their names.

Language file has the following structure:

- In the first line, short name of the used language is given (for example, `en`, `ru`).
- In the second line, name of the used encoding is given (for example, `koi8-r`).
- In the third line, number of bits in CTE is given (`7bit` or `8bit`).
- Other lines contain strings of the `N="text"` type, where `N` – number (identifier) of the string, and `text` – the used text.
- Also a language file can contain empty strings and comments (starting from `#` symbol). These strings are ignored.

Please note that only short names of languages (from the first line of a language file) can be used as a value of the **NotifyLangs** parameter (see above).

Example of a language file:

```
#language name = LANG
en
#coding system = CHARSET
UTF8
#Content-Transfer-Encoding: 7bit/8bit
8bit

1 = "OK"
2 = "password protected, skipped"
...
```

All plug-ins use only the language file specified first in the **NotifyLangs** list for **Notifier**. For searching the necessary string (upon handling of `$n` macro specified, for example, for **add-header action**), the plug-in always uses the following algorithm:

- 1) Directory with language files is determined (value of the **LngBaseDir** parameter from the `[Notifier]` section is always used);
- 2) Used language is determined according to the first value in the **NotifyLangs** list from the `[Notifier]` section;
- 3) Required language file is searched by the following criteria: its name contains prefix with the plug-in name and its first line contains necessary short language name;
- 4) If the file is found, the string with required `n` number is searched. It is assumed that text in the string is encoded with the CTE and encoding that are specified in the file header.

The found string will be used for message processing (added either to the header or message body, depending on plug-in action). Encoding and CTE are specified in the language file header.

If either the required language file or string in the file cannot be found, an error occurs. This error is handled according to the **ProcessingError** parameter value.

If necessary, you can add some strings in the language files. At that, added strings must satisfy the following criteria:

- avoid already occupied numbers, because these strings are used by **Dr.Web MailD** modules (or a plug-in);
- use the encoding and CTE specified in file header.



Plug-Ins

At the current moment, the following plug-ins of **Dr.Web MailD** are available:

- **Drweb** anti-virus plug-in. This plug-in is used for checking email messages for viruses, threats and other malware;
- **Vaderetro** anti-spam plug-in. This plug-in is used for checking email messages for spam attributes;
- **Dr.Web HeadersFilter**. Plug-in which filters email messages by values of their headers;
- **Dr.Web Modifier**. Plug-in which allows modification of email message parts.

Each plug-in is presented as a shared library (a file with `.so` extension). Plug-in libraries exist in `%bin_dir/maild/plugins` directory. Library files of each plug-in has the name of the following pattern, `lib<name>.so`, where `<name>` - name of the plug-in. For example, file of **Drweb** plug-in library is named `libdrweb.so`.

Each plug-in uses its own configuration file. Plug-in configuration files exist in the `%etc_dir` directory. Plug-in configuration files are named according to the following pattern `plugin<name>.conf`, where `<name>` - name of the plug-in. For example, configuration file of **Drweb** plug-in is named `plugin_drweb.conf`.

If necessary, you can configure each plug-in to use configuration files and dynamic libraries, name of which does not correspond to the pattern. To do this, adjust the settings in the of the `[Filters]` section of the main **Dr.Web MailD** configuration file.



On startup, **Dr.Web MailD** temporarily renames files of used plug-in libraries by adding an additional extension `.cache` to the file name. It is done to avoid conflicts when updating libraries by **Dr.Web Updater component**.

For example, **Drweb** plug-in library is renamed on startup to `libdrweb.so.cache`.

Drweb Anti-Virus Plug-In

Drweb is an anti-virus plug-in for **Dr.Web for UNIX mail servers**. It performs anti-virus check of electronic mail.

For proper operation of **Drweb** plug-in, **Dr.Web CoreEngine** and **Dr.Web Daemon** are required – they perform direct anti-virus check. **Dr.Web Daemon** and **Dr.Web CoreEngine** are included into general distribution package of **Dr.Web for UNIX mail servers** and must be installed before installation of the **Drweb** plug-in.

Messages, already segmented, are sent to the `drwebd module` (**Dr.Web Daemon**) for scanning in segments. Therefore, support of MIME processing by **Dr.Web Engine**, as well as by **Dr.Web Daemon**, is not required. Once message analysis is complete, the plug-in sends scanning results to **MailD core** and (if `Yes` is specified as a value of the `AddXHeaders` parameter in the plug-in configuration file) adds the following headers:

- `X-Anti-virus: Name` - where `Name` is the name and version of the anti-virus software;
- `X-Anti-virus-Code` - where `Code` is a termination code of **Dr.Web Daemon**.

Drweb plug-in parameters can be adjusted, by default, in the `plugin_drweb.conf` configuration file.

Connecting Drweb Plug-In

To connect the **Drweb** plug-in to **Dr.Web for UNIX mail servers**, add a `drweb` string (plug-in name)



to the list of plug-ins for message processing in the **Dr.Web MailD** configuration file.

If you want messages to be processed by **Drweb** plug-in before they are imported to the database, add the plug-in name to the list of the **BeforeQueueFilters** parameter values from the [Filter] [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
BeforeQueueFilters = drweb, vaderetro
```

If you want messages to be processed by **Drweb** plug-in after they are imported to the database, add the plug-in name to the list of the **AfterQueueFilters** parameter values from the [Filter] [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
AfterQueueFilters = drweb
```

Setting Drweb Plug-In

All main settings that regulate plug-in operation are set in `%etc_dir/plugin_drweb.conf` configuration file. Description of the configuration file structure and parameter types is provided in [Configuration Files](#). Parameters are described in the order they appear in the main configuration file.

In the [Antivirus] section, general settings for the **Drweb** plug-in are specified:

[Antivirus] section

<pre>Address = {address}</pre>	<p>Socket for interaction between anti-virus plug-in and Dr.Web Daemon.</p> <p>It is possible to specify several sockets for interaction with Dr.Web Daemons that are located on different servers. At that, feature of balancing load on the used servers is enabled.</p> <p>Addresses are listed in the following format: ADDRESS1 [WEIGHT1], ADDRESS2 [WEIGHT2] ...</p> <p>where ADDRESS is a socket address (in the standard form) and WEIGHT is an optional parameter defining priority of this Dr.Web Daemon instance (can be from 0 to 100 inclusive). At least one address in the list must be correct and accessible.</p> <p>Apart from standard address types, you can specify path to the PID file of Dr.Web Daemon, from which necessary information about the sockets can be retrieved.</p> <p>Examples:</p> <p>Specifying path to the PID file: Address = pid:%var_dir/run/drwebd.pid</p> <p>Specifying several addresses: Address = pid:%var_dir/run/drwebd.pid 10, inet:3000@srv2.example.com 5</p> <p>Default value: Address = pid:%var_dir/run/drwebd.pid</p>
<pre>Timeout = {time}</pre>	<p>Timeout for Dr.Web Daemon to execute a command.</p> <p>When the parameter value is set to 0, time is not limited.</p> <p>Default value: Timeout = 30s</p>



<p>ScanType = {local remote auto}</p>	<p>Mode of interaction with Dr.Web Daemon for scanning of email messages. The following modes are allowed:</p> <ul style="list-style-type: none">• local – only names of files to scan are transferred;• remote – only file content is transferred;• auto – Automatic mode. Either file names or file content are transferred. The mode is selected according to the message size, whether local or remote Dr.Web Daemon is used, and mode (synchronous or asynchronous) of message processing by the plug-in. <p>It is strongly recommended to use the auto mode that is specified by default.</p> <p>The local mode can be used only if the scanning Dr.Web Daemon operates on the local host (it is determined by the address type specified in the Address parameter). If at least one of the addresses is remote, it is not recommended to set ScanType=local.</p> <p>Important! If the ScanType parameter has local or auto value, then setting ScanFiles = ByType in the Dr.Web Daemon settings causes Dr.Web Daemon to pass email messages without any check!</p> <p>Default value: ScanType = auto</p>
<p>HeuristicAnalysis = {logical}</p>	<p>R Heuristic analyzer allows Dr.Web Daemon to detect unknown viruses.</p> <p>When Heuristic analyzer is disabled, only known viruses (information on which is stored in virus databases) is detected. Enabling of Heuristic analyzer can result in emergence of false alarms because of the similarity between operation of a legitimate program and virus activity.</p> <p>Usage of Heuristic analyzer can also slightly increase scan time.</p> <p>Default value: HeuristicAnalysis = Yes</p>
<p>TCP_NODELAY = {logical}</p>	<p>If the values is set to Yes, a socket with the enabled TCP_NODELAY parameter will be created.</p> <p>Do not change the default parameter value (No) if you do not have network problems.</p> <p>Default value: TCP_NODELAY = No</p>
<p>ReportMaxSize = {size}</p>	<p>R Maximum size of Dr.Web Daemon log file.</p> <p>When ReportMaxSize = 0, log file size is not limited.</p> <p>It is not recommended to set the parameter value to 0, otherwise log file size can exceed several Mbytes after detection of malware or mail bombs in messages.</p> <p>Default value: ReportMaxSize = 50k</p>
<p>AddXHeaders = {logical}</p>	<p>R If the value is set to Yes, X-Anti-Virus and X-Anti-Virus-Code headers are added to scanned messages.</p> <p>Default value: AddXHeaders = Yes</p>



<p>Paranoid = {logical}</p>	<p>R If Yes value is specified, messages are scanned in the paranoid mode.</p> <p>With this mode enabled, messages are sent to Dr.Web Daemon segment by segment as well as all-in-one-piece. Such strategy allows to increase efficiency of virus detection, but it also increases scan time.</p> <p>Please note, that if a message contains an object to which action pass is applied, then duplication of statistical information on this object may occur (if a virus is detected when processing the attachment or the message itself). Besides, some additional actions (notify, redirect) may be applied twice.</p> <p>Default value: Paranoid = No</p>
<p>RegexsForCheckedFilename = {list of regular expressions}</p>	<p>R A List of regular expressions, used by an anti-virus plug-in to check file names in a report provided by Dr.Web Daemon after a message scan.</p> <p>Names of archived files start with the ">" symbol (number of ">" symbols depends on the archive nesting level). If any part of a file name matches a regular expression from the list, the action specified in the BlockByFilename parameter settings is applied.</p> <p>This check is performed only to files, where no viruses are found.</p> <p>Default value: RegexsForCheckedFilename =</p>
<p>LicenseLimit = {actions}</p>	<p>R Actions to be applied to messages which were not scanned by Dr.Web Daemon due to license expiration.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, tempfail, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p>Default value: LicenseLimit = pass</p>
<p>Infected = {actions}</p>	<p>R Actions to be applied to messages, infected with a known virus.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: cure, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify.</p> <p>Default value: Infected = cure, quarantine</p>
<p>Suspicious = {actions}</p>	<p>R Actions to be applied to messages which could be infected with an unknown virus.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p>



		<p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Suspicious = reject, quarantine, notify</p>
Incurable = {actions}	R	<p>Actions to be applied to incurable messages.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Incurable = reject, quarantine, notify</p>
Adware = {actions}	R	<p>Actions to be applied to messages containing adware.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Adware = reject, quarantine, notify</p>
Dialers = {actions}	R	<p>Actions to be applied to messages containing dialers.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Dialers = reject, quarantine, notify</p>
Jokes = {actions}	R	<p>Actions to be applied to messages containing jokes, which can scare or annoy users.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p>Please note, that several values may be specified at one time.</p> <p><u>Default value:</u> Jokes = reject, quarantine, notify</p>
Riskware = {actions}	R	<p>Actions to be applied to messages containing riskware.</p> <p>In addition to one mandatory action, you can specify several</p>



	<p>optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Riskware = reject, quarantine, notify</p>
<p>Hacktools = {actions}</p>	<p>R Actions to be applied to messages containing programs used to gain unauthorized access to computer systems.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> Hacktools = reject, quarantine, notify</p>
<p>SkipObject = {actions}</p>	<p>R Actions to be applied to messages containing objects which cannot be scanned by Dr.Web Daemon due to the following reasons:</p> <ul style="list-style-type: none">• attachment includes a password-protected or corrupted archive, a symbolic link, a file in a nonstandard format, or an encrypted file• message scan is aborted due to timeout (for details, refer to the description of the SocketTimeout and FileTimeout parameters in the main configuration file <code>drweb32.ini</code>). <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> SkipObject = pass</p>
<p>ArchiveRestriction = {actions}</p>	<p>R Actions to be applied to messages containing archives which cannot be scanned by Dr.Web Daemon due to any of the following restriction exceedings:</p> <ul style="list-style-type: none">• archive compression ratio exceeds the MaxCompressionRatio parameter value• size of packed object exceeds the MaxFileSizeToExtract parameter value• archive nesting level exceeds the MaxArchiveLevel parameter value. <p>All these restrictions are defined in Dr.Web Daemon settings.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject.</p>



	<p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> ArchiveRestriction = reject, quarantine, notify</p>
<p>ScanningErrors = {actions}</p>	<p>R Actions to be applied to messages causing Dr.Web Daemon errors during scan.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, remove, discard, reject, tempfail.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> ScanningErrors = reject, quarantine</p>
<p>ProcessingErrors = {actions}</p>	<p>R Actions to be applied to messages causing Dr.Web Daemon errors during scan.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, discard, reject, tempfail.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p><u>Default value:</u> ProcessingErrors = reject</p>
<p>BlockByFilename = {actions}</p>	<p>R Actions to be applied when one of regular expressions from the RegexsForCheckedFilename parameter matches any file name in Dr.Web Daemon report.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, discard, reject, tempfail.</p> <p>Optional actions are: quarantine, redirect, notify, add-header, score.</p> <p>Please note that when communication with Dr.Web Daemon is performed via the TCP socket, a different format of file names is used in reports.</p> <p>Example: 127.0.0.1 [17078] >/var/drweb/msgs/db/6/00007976/.msg/1.part - Ok</p> <p>That is, they do not start with the ">" symbol, but with an IP address and the number of the scanning process. So, regular expressions in the value of the RegexsForCheckedFilename parameter must be created with consideration of this difference.</p> <p><u>Default value:</u> BlockByFilename = reject, quarantine, notify</p>



When a message is blocked (`reject`) by **Drweb** anti-virus plug-in in the [synchronous mode](#), **Dr.Web MailD** response to a client contains SMTP code (55* or 250, depending on the `ReturnReject` parameter value in the [\[Receiver\] section](#)) and a text message which content is determined by values of the parameters described below. Their values must be enclosed in quotation marks.

<code>UseCustomReply = {logical}</code>	<p>R Use custom messages as an SMTP reply if a message is rejected.</p> <p>Default value: <code>UseCustomReply = No</code></p>
<code>ReplyInfected = {text value}</code>	<p>R Custom message used as an SMTP reply when Infected = <code>reject</code> or Incurable = <code>reject</code> actions are applied, and also when UseCustomReply = <code>yes</code>.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: <code>ReplyInfected = "DrWEB anti-virus: Message is rejected because it contains a virus."</code></p>
<code>ReplyMalware = {text value}</code>	<p>R Custom message used as an SMTP reply when Adware, Dialers, Jokes, Riskware, Hacktools = <code>reject</code> actions are applied and also when UseCustomReply = <code>Yes</code>.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: <code>ReplyMalware = "DrWEB anti-virus: Message is rejected because it contains a malware."</code></p>
<code>ReplySuspicious = {text value}</code>	<p>R Custom message used as an SMTP reply when Suspicious = <code>reject</code> action is applied, and also when UseCustomReply = <code>Yes</code>.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: <code>ReplySuspicious = "DrWEB anti-virus: Message is rejected because it contains suspicious content."</code></p>
<code>ReplySkipObject = {text value}</code>	<p>R Custom message used as an SMTP reply when SkipObject = <code>reject</code> action is applied, and also when UseCustomReply = <code>Yes</code>.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: <code>ReplySkipObject = "DrWEB anti-virus: Message is rejected because it cannot be checked."</code></p>



<pre>ReplyArchiveRestriction = {text value}</pre>	<p>R Custom message used as an SMTP reply when ArchiveRestriction = reject action is applied, and also when UseCustomReply = Yes.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: ReplyArchiveRestriction = "DrWEB anti-virus: Message is rejected because it contains archive which violates restrictions."</p>
<pre>ReplyError = {text value}</pre>	<p>R Custom message used as an SMTP reply when one of the following actions are applied: ScanningErrors, ProcessingErrors, and also when UseCustomReply = Yes.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: ReplyError = "DrWEB anti-virus: Message is rejected due to software error."</p>
<pre>ReplyBlockByFilename = {text value}</pre>	<p>R Custom message used as an SMTP reply when BlockByFilename = reject action is applied, and also when UseCustomReply = Yes.</p> <p>You can specify only the text part of the message. Text must be quoted if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: ReplyBlockByFilename = "DrWEB MailD: Message is rejected due to filename pattern"</p>

If **UseCustomReply** = No or the corresponding string is not specified, the following standard message outputs: "The message has been rejected by the Dr.Web MailD".

Examples

The following examples shows how to set actions of the **Drweb** anti-virus plug-in:

1. **Example of setting Suspicious action.** If a message is infected by an unknown suspicious object, it is necessary to reject the message, move it to **Quarantine**, and add a new header **X-DRWEB-PLUGIN-CHECK-STATUS** which value is extracted from the string with ID 120 in the [default language file](#):

```
Suspicious = reject, quarantine, notify, add-header (X-DRWEB-PLUGIN-CHECK-STATUS:$120)
```

2. **Example of setting ArchiveRestriction action.** If a message contains an attached archive and this archive violates restrictions specified for archive checking, the message must be passed to the recipient (or checked by the rest of the plug-ins), but it is also necessary to increase the message score by 100 points and notify the administrator on that:

```
ArchiveRestriction = pass, notify, score (100)
```



Vaderetro Anti-Spam Plug-In

Vaderetro is a plug-in used in **Dr.Web for UNIX mail servers**. It filters out spam using **VadeRetro** library, designed by **Vade Retro Technology** company (a division of **GoTo Software** company).

VadeRetro library analyzes mail in the autonomous mode without requesting external sources for additional information on spam. Moreover, the library assures high processing speed and constantly improving quality of message analysis, which is possible due to dynamic updates of the library code (through **Dr.Web Updater component**).

File of **VadeRetro** dynamic library, which is used by **Vaderetro** anti-spam plug-in, is located in the `%var_dir/lib` directory and named as `libvaderetro.so` (as well as a file of the plug-in dynamic library). Please note that both files (of **VadeRetro** library and **Vaderetro** plug-in) are named equally, but they are different libraries. File of the **Vaderetro** plug-in dynamic library is located in the `%bin_dir/maild/plugins` directory.



On startup, **Dr.Web MailD** temporarily renames file of **VadeRetro** library by adding a `.cache` extension to the file name. It is done to avoid update conflict when updating **VadeRetro** library through **Dr.Web Updater component**.

Depending on the analysis results, each message processed by the **VadeRetro** library receives a score - an integer in the range from -10000 to +10000. The less is the value, the higher is the probability that the message is not spam. Threshold value, which determines whether to classify a message as spam, is defined by the `SpamThreshold` parameter from the `vaderetro` configuration file. If the evaluation score given to a message is higher than or equal to the `SpamThreshold` parameter value, the message is classified as spam.

At the final stage of analysis **VadeRetro** library can add the following headers to the message:

- `X-Drweb-SpamScore: n`, where `n` is the score that **VadeRetro** assigns to a message. The header is added only if the `AddXHeaders` parameter value is set to **Yes**.
- `X-Drweb-SpamState: b`, where `b` is **Yes** for spam and infected messages and **No** for non-spam messages and DSN. The header is added only if the `AddXHeaders` parameter value is set to **Yes**.
- `X-Drweb-SpamState-Num: s`, where `s` is a message classification result; `s` can take the following values: 0, 1, 2 and 3.
 - `s = 0` – this message is not spam;
 - `s = 1` – this message is spam;
 - `s = 2` – this message contains a virus;
 - `s = 3` – this message is a DSN.

This header is added only if **Yes** is specified for the `AddXDrwebSpamStateNumHeader` parameter of the `vaderetro` configuration file.

- `X-Drweb-SpamVersion: version`, where `version` is the version of **VadeRetro** library. This header is added only if **Yes** is specified for the `AddVersionHeader` parameter of the **Vaderetro** configuration file.
- `X-Spam-Level: z`, where `z` is a set of "*" (each of them equals to 10 score points, assigned to a message). This header is added only if **Yes** is specified for the `AddXSpamLevel` parameter of the **Vaderetro** configuration file.
- `X-DrWeb-SpamReason: some_text`, where `some_text` is some encoded diagnostic message from the anti-spam module. It is necessary for improvement of the quality of spam detection. This header is added only when the `AddXHeaders` parameter for this message is set to **yes**.



Moreover, if a message is classified as spam because its score is equal or less than the **Threshold** value, **Vaderetro** plug-in can add the text specified as the **SubjectPrefix** parameter value in the **Vaderetro** configuration file to the message subject. The text is added only if the **SubjectPrefix** parameter value is not an empty string.

Similarly to that, the **NotifySubjectPrefix** parameter value can be added to the beginning of the **Subject** field of a notification.

If a message was marked as unconditional spam according to the **UnconditionalSpamThreshold** parameter, a value of **UnconditionalSubjectPrefix** is added to the beginning of the **Subject** field of this message.

A message score can change depending on the information on the sender and recipient addresses:

1. You can specify white and black lists of senders' addresses (**WhiteList** and **BlackList** configuration parameters respectively). If one of the senders' addresses is in the black or white list, the message score is changed by 5000 points (increased or decreased respectively) for every address found in the list. For details, refer to the [description of the parameters](#).
2. You can specify number of points by which it is required to change the score of a message from protected networks (that is, networks specified in the **ProtectedNetworks** parameter of the [Maild] [section](#) in the main **Dr.Web MailD** configuration file).
3. You can also use special cache `reply_cache`, that stores information on messages from protected networks (list of recipient addresses) in order to consider this information while analyzing messages sent to the protected networks and being a reply on the messages. If the message sender is already cached, the message score can be changed by the specified number of points.

Note that a message undergoes all checks successively, so if several conditions were true for the same message, all changes are summarized. For example, if a message sender is in the black list and the sender is in the `reply_cache`, the message score is increased by a penalty for the sender being in the black list and then by a value specified in the **ReplyToProtectedNetworkScoreAdd** plug-in [parameter](#) of the plug-in configuration.

Messages which were mistakenly marked as spam should be sent to vrnospam@drweb.com. Spam messages, accidentally passed by the spam filter, should be sent to vrspam@drweb.com.

Installing Vaderetro Plug-In

To connect **Vaderetro** plug-in to **Dr.Web for UNIX mail servers**, add `vaderetro` string to the list of plug-ins for message processing in the **Dr.Web MailD** configuration file.

If you want messages to be processed by **Vaderetro** plug-in before they are imported to the database, add the plug-in name to the list of the **BeforeQueueFilters** parameter values from the [Filter] [section](#) of **Dr.Web MailD** configuration file.

Example:

```
BeforeQueueFilters = drweb, vaderetro
```

If you want messages to be processed by the **Vaderetro** plug-in after they are imported to the database, add the plug-in name to the list of the **AfterQueueFilters** parameter values from the [Filter] [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
AfterQueueFilters = vaderetro
```



Setting Vaderetro Plug-In

All main parameters that regulate plug-in operation are set in `%etc_dir/plugin_vaderetro.conf` configuration file. Description of the configuration file structure and parameter types is provided in [Configuration Files](#). Parameters are described in the order they appear in the main configuration file.

In the `[VadeRetro]` section, general settings for `vaderetro` plug-in are specified:

[Vaderetro] section

<pre>PathToVadeRetro = {path to file}</pre>	<p>Path to the VadeRetro anti-spam library.</p> <p>It is possible to enable dynamic updates with Dr.Web Updater component. It will download a new library version, replace the old library with it and send <code>SIGHUP</code> signal to <code>drweb-maild</code> process.</p> <p><u>Default value:</u> PathToVadeRetro = <code>%var_dir/lib/libvaderetro.so</code></p>
<pre>FullCheck = {logical}</pre>	<p>R The parameter determines strategy of message check for spam. If the parameter is set to <code>No</code>, a message is checked for spam only if the total sum of "positive" message scores (for example, scores given is the message sender is specified in the white list) is under the threshold set in the <code>VadeRetro</code> library. Otherwise (if the parameter value is set to <code>Yes</code>), a message is checked for all spam attributes, regardless of the total value of its "positive" scores.</p> <p>Please note that enabled full check (when <code>Fullcheck = Yes</code>) can slow down the overall operation speed.</p> <p>Also if this parameter value is set to <code>Yes</code>, specifying the message sender in the white list (see below) might have no effect: the message can be classified as spam based on the analysis of its inner content, even if its sender belongs to the white list.</p> <p><u>Default value:</u> FullCheck = <code>Yes</code></p>
<pre>NoHamFrom = {logical}</pre>	<p>R <code>Yes</code> value of this parameter disables check of messages sent to the addresses specified in the embedded <code>ham</code> list of <code>VadeRetro</code> library (for example, <code>nospam@domain.ru</code>).</p> <p>Note that the list is embedded and cannot be modified.</p> <p><u>Default value:</u> NoHamFrom = <code>Yes</code></p>
<pre>AddXHeaders = {logical}</pre>	<p>R Add <code>X-Drweb-SpamState</code> and <code>X-Drweb-SpamScore</code> headers to a message.</p> <p>The first one contains information whether or not a message is spam. The second one contains the total message score after full check.</p> <p><u>Default value:</u> AddXHeaders = <code>Yes</code></p>
<pre>AddVersionHeader = {logical}</pre>	<p>R Add <code>X-Drweb-SpamVersion</code> header with information on Vaderetro version.</p> <p><u>Default value:</u> AddVersionHeader = <code>No</code></p>
	<p>R Add <code>X-Drweb-SpamState-Num</code> header to a message.</p>



<code>AddXDrwebSpamStateNumHeader = {logical}</code>		<p>It includes numerical value assigned by the VadeRetro library according to the classification results (a message can be classified as one of the following):</p> <ul style="list-style-type: none">• 0 - this message is not spam;• 1 - this message is spam;• 2 - this message is infected with virus;• 3 - this is DSN. <p><u>Default value:</u> AddXDrwebSpamStateNumHeader = No</p>
<code>AddXSpamLevel = {logical}</code>	R	<p>Add X-Spam-Level header to a message. It consists of * symbols.</p> <p>Each * symbol means 10 score points.</p> <p>For example, message with 110 score points will get X-Spam-Level: ***** header.</p> <p><u>Default value:</u> AddXSpamLevel = No</p>
<code>CheckForViruses = {logical}</code>	R	<p>Enable heuristic check for viruses by the VadeRetro library.</p> <p>Upon detection of a virus, the library classifies the message to group 2.</p> <p><u>Default value:</u> CheckForViruses = Yes</p>
<code>CheckDelivery = {logical}</code>	R	<p>Enable spam check for those messages that are classified by the VadeRetro library as DSN (messages of group 3).</p> <p>If spam attributes are found, the message is classified to group 1.</p> <p><u>Default value:</u> CheckDelivery = No</p>
<code>AllowRussian = {logical}</code>	R	<p>Determines whether to add extra scores to a message with Cyrillic text (if the value is set to Yes, the scores are not added).</p> <p><u>Default value:</u> AllowRussian = Yes</p>
<code>AllowCJK = {logical}</code>	R	<p>Determines whether to add extra scores to a message with Chinese, Japanese or Korean text or not (if the value is set to Yes, the scores are not added).</p> <p><u>Default value:</u> AllowCJK = Yes</p>
<code>WhiteList = {LookupLite}</code>	RA	<p>White list of sender addresses.</p> <p>Sender addresses are taken from <code>Return-Path</code> and <code>From</code> headers of an email message. If an email message does not contain the headers or there is one or more empty strings before the <code>From</code> field, then sender address is not searched in the white list. If a message contains more than one <code>From</code> field, the address is taken from the first field.</p> <p>It is allowed to use search templates (wildcards) <code>*@<domain></code> as elements of the list. For example, <code>*@mycompany.com</code> string matches all addresses from domain <code>mycompany.com</code>.</p> <p>Specified domains must be FQDN.</p>



If the sender's address specified in the `From` field is in the white list, the message score is decreased by 5 000 points. If the addresses specified in the both fields (`From` and `Return-Path`) are from the white list, the message score is decreased by 10 000 points

Please note the following features of white list processing:

1. White list is not sorted, so it is possible that the same address is accidentally repeated in the list. In this case, the total message score is decreased by 5 000 points each time, the address (from the `Return-Path` and `From` fields) is specified in the list (for example, if the address is found 3 times in the list – the score will be decreased by 15 000 points).
2. If the sender's domain (from the `Return-Path` and `From` fields) equals to the receiver's domain (from the `To:` field), and this domain is specified in the white list as a wildcard (`*@<domain>`), then the sender's address is not checked (and the message score is not changed). The same behaviour happens if the sender's and recipient's addresses are equal and specified in the white list.
3. If the sender's domain `<domain1>` is not equal to the recipient's domain `<domain2>` and both of the domains are presented in the white list as wildcards (`*@<domain1>` and `*@<domain2>` respectively), then the sender's address is checked, and the message score is decreased by 10 000 points. The same behaviour happens if the sender's and recipient's addresses are different and both are specified in the white list.

Please note that if `FullCheck = Yes`, this parameter **might have no effect**: if a message is considered spam according to its content analysis results, the scores assigned by this parameter are ignored and will not be subtracted from the total message score (see above)!

Please note that the parameter value is [LookupLite](#).

Example:

```
hello@myneighbourhood.co.uk
*@mycompany.com
```

Default value:

`WhiteList =`

`BlackList =`
{`LookupLite`}



Black list of senders.

Similar to the `WhiteList` parameter.

Specified domains must be FQDN.

If the sender's address from is found in the black list, the message score is increased by 5 000 points. If the addresses from both fields (`Return-Path` and `From`) are specified in the black list, the message score is increased by 10 000 points.

Please note the following features of black list processing:

1. Black list is not sorted, so it is possible that the same address is accidentally repeated in the list. In this case, the total message score is increased by 5 000 points each time the address (from the `Return-Path` and `From` fields) is specified in the list (for example, if the address is found 3 times in the list – the score will be increased by 15 000 points).
2. If the sender's domain equals to the recipient's domain and the domain is specified in the black list as a wildcard (`*@<domain>`), the sender address **is checked** (and message score **is changed**). The same behaviour happens if the sender's address equals to the recipient's address and is specified in the black list.

Please note that the parameter value is [LookupLite](#).



	<p>Default value: BlackList =</p>
<p>SpamThreshold = {числовое значение}</p>	<p>The parameter value is a threshold for the total message score. If the score of a message is greater than or equal to the specified threshold, the message is identified as spam by Vaderetro plug-in and X-Drweb-SpamState header is set to Yes.</p> <p>If the message score is greater than or equal to the SpamThreshold parameter value, but is less than the value of the UnconditionalSpamThreshold parameter (see below), the action specified in the Action parameter is applied and the text specified in the SubjectPrefix parameter is added to the message subject (see below).</p> <p>SpamThreshold parameter value must be less than or equal to the value of the UnconditionalSpamThreshold parameter.</p> <p>Please note that Vaderetro plug-in classifies a message as spam or not spam only according to the ratio of the message score to the SpamThreshold value. At that, the class to which Vaderetro classified the message is not taken into account. For example, a message can be considered as spam by the library and classified to group 1, but if its score, assigned by the library, is less than the specified threshold, the message is not indicated as spam by Vaderetro plug-in (X-Drweb-SpamState header is set to No) and thus, the action specified for spam messages is not applied to it.</p> <p>Default value: SpamThreshold = 100</p>
<p>UnconditionalSpamThreshold = {numerical value}</p>	<p>R The parameter value is a threshold for the total message score. If a message score is greater than or equal to this parameter value, the message is identified as unconditional spam and X-Drweb-SpamState header is set to Yes.</p> <p>In this case, action specified in the UnconditionalAction parameter is applied to the message and the text specified in the UnconditionalSubjectPrefix parameter is added to the message subject (see below).</p> <p>Value specified in the UnconditionalSpamThreshold parameter must be greater than or equal to the value of the SpamThreshold parameter.</p> <p>Default value: UnconditionalSpamThreshold = 1000</p>
<p>Action = {actions}</p>	<p>R Actions to be applied to a message which was identified as spam by Vaderetro plug-in.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, reject, discard, tempfail.</p> <p>Additional actions are: quarantine, redirect, add-header, score.</p> <p>Default value: Action = pass</p>
<p>UnconditionalAction = {actions}</p>	<p>R Actions to be applied to a message which was identified as unconditional spam by Vaderetro plug-in.</p>



	<p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are pass, reject, discard, tempfail.</p> <p>Optional actions are: quarantine, redirect, add-header, score.</p> <p><u>Default value:</u> UnconditionalAction = pass</p>
<p>NotifyAction = {actions}</p>	<p>R Actions applied to a message which was identified as spam or unconditional spam by Vaderetro plug-in according to the message score, and, moreover, classified as DSN (message of group 3) by the VadeRetro library.</p> <p>In addition to one mandatory action, you can specify several optional actions.</p> <p>Mandatory actions are: pass, reject, discard, tempfail.</p> <p>Additional actions are: quarantine, redirect, add-header, score.</p> <p><u>Default value:</u> NotifyAction = pass</p>
<p>SubjectPrefix = {text}</p>	<p>R Prefix added to the message subject, if the message is identified as spam by Vaderetro plug-in according to the message score (the message score must be greater than or equal to the SpamThreshold parameter value).</p> <p>See also a note below the table.</p> <p><u>Default value:</u> SubjectPrefix =</p>
<p>UnconditionalSubjectPrefix = {text}</p>	<p>R Prefix added to the message subject, if the message is identified as unconditional spam by Vaderetro plug-in according to the message score (the message score must be greater than or equal to the UnconditionalSpamThreshold parameter value).</p> <p>It is added, when a message score is greater than the UnconditionalSpamThreshold parameter value.</p> <p>See also a note below the table.</p> <p><u>Default value:</u> UnconditionalSubjectPrefix =</p>
<p>NotifySubjectPrefix = {text}</p>	<p>R Prefix added to the message subject, if the message is identified as spam or unconditional spam by Vaderetro plug-in according to the message score, and, moreover, classified as DSN (message of group 3) by the VadeRetro library.</p> <p>See also a note below the table.</p> <p><u>Default value:</u> NotifySubjectPrefix =</p>
<p>FromProtectedNetworkScoreAdd = {numerical value}</p>	<p>R If a message is sent from a protected network (specified in the ProtectedNetworks list in the [MailD] section of the main Dr.Web MailD configuration file), the message score increases by the specified value (the value may be negative).</p>



	<p>If you want to disable this function, specify 0 as a value of this parameter.</p> <p><u>Default value:</u> FromProtectedNetworkScoreAdd =</p>
<pre>UseReplyCache = {Yes No}</pre>	<p>Manages the ProtectedNetworkReplyCacheLifeTime and ReplyToProtectedNetworkScoreAdd parameters (enables and disables use of <code>reply_cache</code>).</p> <p>This cache is used as temporary storage of the message data (addresses of all its recipients) to consider the data while checking the message for spam. It is to be sent in reverse direction in reply to the checked messages (with the Reply-to header).</p> <p>If the value is set to Yes, reply_cache storage is used.</p> <p><u>Default value:</u> UseReplyCache =</p>
<pre>ProtectedNetworkReplyCacheLifeTime = {time}</pre>	<p>R Time period to store data on a message in <code>reply_cache</code> if the sender's address is in the ProtectedNetworks list ([Maild] section of the main Dr.Web MailD configuration file).</p> <p>If an added address is already in <code>reply_cache</code>, the entry is updated.</p> <p><u>Default value:</u> ProtectedNetworkReplyCacheLifeTime =</p>
<pre>ReplyToProtectedNetworkScoreAdd = {numerical value}</pre>	<p>R Value that must be added to the current message score if the message sender is in reply_cache.</p> <p>The added value can be negative (to decrease the score). If you want to disable this function, specify 0 as a value of this parameter. Moreover, if <code>reply_cache</code> is disabled (that is, UseReplyCache = No), the sender will never be found there and any value specified for this parameter, in fact, is not used.</p> <p><u>Default value:</u> ReplyToProtectedNetworkScoreAdd =</p>

When a message is blocked (`reject`) by the plug-in in the [synchronous mode](#), **Dr.Web MailD** response to a client contains SMTP code (55* or 250, depending on the **ReturnReject** parameter value in the [Receiver] [section](#)) and a text message which content is determined by values of the parameters described below. Their values must be enclosed in quotation marks.

<pre>UseCustomReply = {logical}</pre>	<p>R Use custom messages as an SMTP reply when messages are rejected.</p> <p><u>Default value:</u> UseCustomReply = No</p>
<pre>SpamCustomReply = {text}</pre>	<p>R Custom message used as an SMTP reply when the following actions are applied Action, UnconditionalAction, NotifyAction = <code>reject</code> actions and also when UseCustomReply = <code>yes</code>.</p> <p>You can specify only the text part of the message. Text must be enclosed in quotation marks if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p>



Default value:

```
SpamCustomReply = "Dr.Web vaderetro plugin: this is spam!"
```

If `UseCustomReply` = `No` or the corresponding string is not specified, the following standard message outputs: "The message has been rejected by the Dr.Web MailD".



Note that it is required to **consider encoding of the search text** for **all parameters** that change or search a header value or a part of a message. Rules of working with header values in custom encoding are described in the [Work with String Values](#).

Examples

Examples of **Vaderetro** anti-spam plug-in usage.

1. If an email message score exceeds the `Threshold` value:

- This message is blocked (client gets 550 SMTP reply code);
- Message is moved to **Quarantine**;
- Copy of the message is sent to the address specified in the value of the `AdminMail` parameter in the `[Notifier]` [section](#) (**Dr.Web MailD configuration file**).

```
Action = reject, quarantine, redirect
```

2. Similarly to the first example, but copies of the message are to be sent only to the specified addresses (to the address set as the `AdminMail` parameter (`[Notifier]` [section](#)) value nothing is sent):

```
Action = reject, quarantine, redirect (admin1@domain | admin2@domain | admin3@domain)
```

Dr.Web HeadersFilter Plug-In

Dr.Web HeadersFilter plug-in filters messages according to their headers. When filtration rules are set, regular expressions (**Perl** syntax) can be used.

Connecting Headersfilter Plug-In

To connect **Dr.Web HeadersFilter** plug-in to **Dr.Web for UNIX mail servers**, in the **Dr.Web MailD** configuration file add `headersfilter` string to the list of plug-ins which process messages.

If you want messages to be processed by **Dr.Web HeadersFilter** before they are moved to the database, you must add the name of this plug-in to the list of the `BeforeQueueFilters` parameter values from the `[Filter]` [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
BeforeQueueFilters = drweb, headersfilter
```

If you want messages to be processed by **Dr.Web HeadersFilter** after they are moved to the database, you must add the name of this plug-in to the list of the `AfterQueueFilters` parameter values from the `[Filter]` [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
AfterQueueFilters = headersfilter
```



Setting Headersfilter Plug-In

All main parameters that regulate plug-in operation are set in `%etc_dir/plugin_headersfilter.conf` configuration file. Description of the configuration file structure and parameter types is provided in [Configuration Files](#). Parameters are described in the order they appear in the main configuration file.

In the `[Headersfilter]` section, general settings for **Dr.Web HeadersFilter** plug-in are specified.

Filtration parameters are defined by the **filter rules** which are described below. Rules are analyzed in the same order as they are listed in the section, that is a rule which is set first in the list is analyzed first. Rules are analyzed until a suitable rule is found and the plug-in executes the action which is set for this rule.

If **Reject*** filter rule is applied to a message, the message is not processed further. If **Accept*** filter rule is applied to a message, other rules are ignored and the message is processed by other plug-ins of **Dr.Web MailD** (if some plug-ins did not process the message).

[headersfilter] section

<pre>ScanEncodedHeaders = {logical}</pre>	<p>R Scan headers before decoding.</p> <p>For example, <code>Yes</code> value for the <code>ScanEncodedHeaders</code> parameter and</p> <pre>RejectCondition = Subject = "iso-8859-5"</pre> <p>condition allow to filter out messages, <code>Subject</code> field of which is encoded with <code>iso-8859-5</code>.</p> <p>Please note that if <code>Yes</code> value is specified, all encoded headers are scanned twice: before and after decoding (and scanning stops if any of the rules is activated).</p> <p><u>Default value:</u></p> <pre>ScanEncodedHeaders = Yes</pre>
<pre>RejectCondition = {conditions}</pre>	<p>R A Message filtering rules.</p> <p>If a message header matches any specified condition, the message is filtered out. Actions to be applied for filtered messages can be specified in the <code>Action</code> parameter of this section. The rules can be specified for any header.</p> <p>Ordinarily, a rule consists of a header name and a regular expression:</p> <pre>HEADER = regular_expression</pre> <p>You can combine several conditions using brackets or logical operators <code>OR</code> and <code>AND</code>. The <code>"!="</code> (not equal) operator can also be used. Expressions containing white spaces must be enclosed in quotation marks.</p> <p>Example:</p> <pre>RejectCondition = Subject = "money" AND Content-Type = "text/html"</pre> <p>Moreover, there are two additional types of filtration:</p> <ul style="list-style-type: none">• <code>No HEADER_NAME</code> - condition that allows filtration of messages without a certain header. <p>Example:</p> <pre>RejectCondition = No From</pre> <p>– filters out all messages without the <code>From</code> field.</p> <ul style="list-style-type: none">• <code>HEADER_NAME = "8bit"</code> - filters out all messages with



		headers containing 8-bit symbols. See also a note below the table.
		<u>Default value:</u> RejectCondition =
AcceptCondition = {conditions}	RA	Rules for accepting messages. If a message header matches any specified condition, scanning stops, and message is immediately sent to other plug-ins for further processing. Acceptance conditions can be specified for any header. Description of condition set for the RejectCondition parameter is also true for the AcceptCondition parameter. Thus, for details on AcceptCondition refer to the RejectCondition parameter description provided above. <u>Default value:</u> AcceptCondition =
FilterParts = {logical}	R	The Yes value enables processing of rules specified by the RejectPartCondition and AcceptPartCondition parameters. <u>Default value:</u> FilterParts = Yes
RejectPartCondition = {conditions} AcceptPartCondition = {conditions}	RA	Filtering rules that are similar to those for the RejectCondition and AcceptCondition parameters, but affecting only headers of attached objects. Also the following condition can be used: FileName = mask, where mask is a regular expression. Filtration of messages according to these rules is possible only if Yes value is specified for the FilterParts parameter. <u>Default value:</u> RejectPartCondition = AcceptPartCondition =
MissingHeader = {text}	RA	Set of missing headers to be used as a filtration condition. Example: MissingHeader = "To", "From" <u>Default value:</u> MissingHeader =
Action = {actions}	R	Actions to be applied to filtered messages. In addition to one mandatory action, you can specify several optional actions. Mandatory actions are: pass, tempfail, discard, reject. Optional actions are: quarantine, redirect, notify, add-header, score. <u>Default value:</u> Action = reject, notify



When a message is blocked (`reject`) by the plug-in in the [synchronous mode](#), **Dr.Web MailD** response to a client contains SMTP code (55* or 250, depending on the `ReturnReject` parameter value in the [\[Receiver\] section](#)) and a text message which content is determined by values of the parameters described below. Their values must be enclosed in quotation marks.

<pre>UseCustomReply = {logical}</pre>	<p>R Use custom messages as an SMTP reply if a message is rejected.</p> <p>Default value: <code>UseCustomReply = No</code></p>
<pre>ReplyRuleFilter = {text}</pre>	<p>R Custom message used as an SMTP reply when <code>Action = reject</code> is applied and also when <code>UseCustomReply = yes</code>. You can specify only the text part of the message. Text must be enclosed in quotation marks if it contains white spaces.</p> <p>Example: 550 5.7.0 "Text part of reply"</p> <p>Default value: <code>ReplyRuleFilter = "DrWEB HeadersFilter plugin: Message is rejected by headers rule filter."</code></p>

If `UseCustomReply = No` or the corresponding string is not specified, the following standard message outputs: "The message has been rejected by the Dr.Web MailD".



Note that it is required to **consider encoding of the search text** for **all filter rules** that change or search a header value or a part of a message. Rules of working with header values in custom encoding are described in the [Work with String Values](#).

Examples

The following examples shows how to set actions of **Dr.Web HeadersFilter** plug-in:

- Suppose that it is required to filter email messages either without the `To` header or `Subject` of which equals to Russian word "rect" (in CP1251 encoding). The filtered messages must be rejected and moved to **Quarantine**. It is also necessary to notify administrator on that:

```
MissingHeader = "To"
RejectCondition = Subject = "=\\?cp1251\\?B\\?8uXx8go=\\?="
Action = reject, quarantine, notify
```

- Suppose that if a message does not contain `Subject`, it is required to pass the message, but add the "(none)" string to the `Subject` field and increase the message score by 500 points:

```
MissingHeader = "Subject"
Action = pass, score (500), add-header ("Subject: (none)")
```

Dr.Web Modifier Plug-In

Plug-in fuctions

Dr.Web Modifier plug-in is used for:

- Mail content analysis* – search of attached objects of certain MIME types (graphic objects, executables, media files) as well as MIME objects that satisfy certain criteria;
- Message body modification* – removing of MIME objects that satisfy certain criteria, modification of headers and/or content of MIME objects;
- Mail processing management* – blocking, moving to **Quarantine**, redirecting, adding headers and



scores depending on MIME objects found in message bodies.

Details on the message structure used while analyzing and processing by **Dr.Web Modifier** is provided below.

Message is a composite object that can be selected to perform an action as a whole or partially, as every message can be presented as a hierarchy set of elements (MIME objects, their headers and content). It is also possible to select a part of a message for individual processing. The following figure shows hierarchy structure of a message with nested objects.

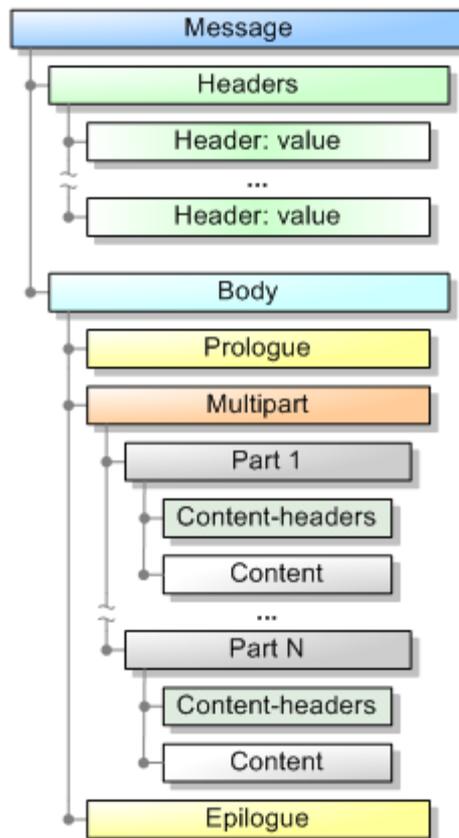


Figure 19. Hierarchy structure of a message with nested objects

Message consists of the following parts:

1. At the top level, a message is divided into **Headers** and **Body**.
2. If the message body content is of `multipart/*` MIME type, the body is a composite object with the following parts:
 - **Prologue** – text inserted into the body before the first nested MIME object. This text is not displayed in MUA supporting MIME multipart and is usually absent;
 - **Epilogue** – text inserted into the body after the last nested MIME object. This text is not displayed in MUA supporting MIME multipart and is usually absent;
 - If the message body consists of several parts, each of these parts is treated as a separate MIME object with its own set of headers (usually they are content headers, such as `Content-type`, `Content-description`, `Content-disposition` and others), its own content and, in some cases, a prologue and epilogue. MIME objects, can also be composite (that is, be of the `multipart/*` type).

Otherwise, a message body is treated as a whole object that does not contain nested parts, a prologue and an epilogue.

As any nested object has the same structure as a message (object also consists of a header, body, and,



in some cases, prologue and epilogue), messages can be treated as a MIME object with headers (**Headers**) and content (**Body**). This MIME object is called a **root** object.

To search objects in a message, **Dr.Web Modifier** supports the following versions of regular expressions:

- basic regular expressions;
- extended regular expressions;
- Perl-compatible regular expressions.

For details on regular expressions, you can visit **Wikipedia** ([Regular expression](#) article).

Order of modification rules for message processing

The plug-in processes messages according to **modification rules**. These rules are divided into *local rules* and *global rules*. At first, local rules are applied, and then - global rules. If a message does not have local modification rules specified for it (that depends on the Processing rules applied for the message, see below), the message is checked only according to global modification rules. If a message does not satisfy local modification rules, it is rejected and check with the use of global rules is not performed.

Local modification rules used by **Dr.Web Modifier** can be specified in [Message processing rules](#) either in the [Rules] [section](#) of the **Dr.Web MailD configuration file** or in the [built-in database](#) of users and user groups. For modification rules, the `modifier/LocalRules` parameter is used. Global modification rules, used by **Dr.Web Modifier**, can be specified only in the `GlobalRules` parameter in the plug-in [configuration file](#) ([Modifier] section).



Note that the `GlobalRules` parameter cannot be used in Message processing Rules and value of the `LocalRules` parameter cannot be specified in the plug-in configuration file.

Modification rule format

Any modification rule consists of the following parts:

```
<Selection operator>, [<Conditional validator>], <Action operator> [, <Action operator>, ...]
```

Note that operators in modification rules are separated by commas. You can set more than one action operators. Conditional validation can be absent. Minimum required form of the rule includes a selection operator and an action operator.

Modification rules are always specified on a single line. If it is necessary to split the rule on several lines, specify the '\ ' symbol at the end of the line immediately before the rule to split, for example:

```
<Selection operator>, \  
<Action operator>, <Action operator>
```

As an action operator, you can also specify an element fetch with its conditional validators and action operators.

1. Selection operators

Selection of items from the message is performed with the `select <element>` operator, where `<element>` is a placeholder for the type of elements to be selected for analysis and processing. The following types are available:

- **message** – select a root MIME object (that is a whole message).



- **mime** (<SEGMENT>) | **mime.**<SEGMENT> [[<name>] <regular_expression>] – select MIME objects or their content from the specified segment.

Difference between syntax with brackets and syntax with a dot is the following: argument in brackets selects actual objects that contain elements in the specified segment, argument with a dot - only elements of these objects (certain headers, text of the prologue, body content and others).

You can set the following values as a <SEGMENT>:

- `headers` – header area of the MIME object;
- `prologue` – prologue area of the MIME object;
- `body` – body area of the MIME object (content);
- `epilogue` – epilogue area of the MIME object.

You can specify the following additional parameters:

- <name> – name of the search header. Specify only if the value of <SEGMENT> is set to `headers`.
- <regular_expression> – template for search (for example, template that matches the header or text of the element).

Example:

The following command selects content of all video fragments from the message:

```
select mime(headers) Content-type "x-video"
```

The following command selects information on type of data from all video fragments (value from the header):

```
select mime.headers Content-type "x-video"
```

In a header (for compatibility with **Vaderetro plug-in**), you can also use instructions to compare with integer ">" and "<" (a backslash '\' is not required to escape the symbols!). The compared header is considered matching the selection criterion if this header contains an integer (for example, `X-Drweb-SpamScore: "30"`) satisfying the condition.

Example:

```
select mime(headers) X-Drweb-SpamScore "<50"
```

This command selects items that contain the `X-Drweb-SpamS` header which value is an integer less than 50. Note that a backslash is not required to escape the "<" symbol. If a modification rule is specified as follows:

```
select mime(headers) X-Drweb-SpamScore "\\<50"
```

elements with `X-Drweb-SpamScore: "<50` header are selected.

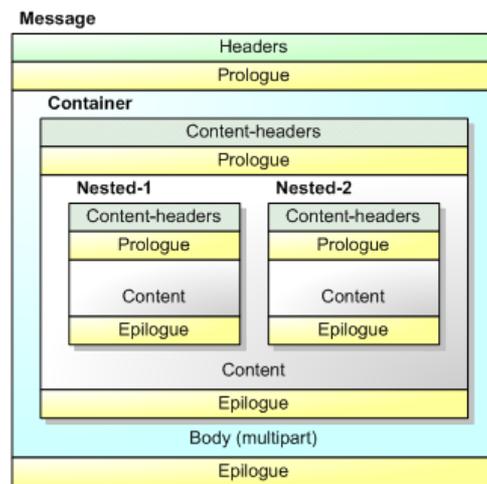


Note that you cannot select a composite MIME object with `select mime (<SEGMENT>)` command except when the object is root (a message).

For example, if a message has the structure as presented in the figure below, the `select mime (headers)` command selects the following objects with headers:

- **Message** (root object "message");
- **Nested-1**;
- **Nested-2**.

Container object is not selected, as it is composite (includes **Nested-1** and **Nested-2** objects) and it is not root (not a message).



- **sender** <regular_expression>, **recipient** <regular_expression> – Selects the whole message if it contains the corresponding record about the sender (recipients). Data about the sender (recipients) is taken from the message envelope. If the source sequence of symbols is found, `select message` command is performed (a whole message is selected).

Example:

If a message is sent to the administrator, you can add a welcome note at the end using the `append_text` command (see below):

```
select recipient "root@localhost", \  
append_text "hello, root"
```

It is clear that if the message is not sent to the administrator, no items are selected and therefore the `append_text` command is not applied. This feature allows avoiding use of conditional operators if the alternative action (when required elements are not found) is absent.

- **select_mimes** – This command allows moving from selection of headers to selection of MIME objects that contain them. That enhances plug-in operation when it is required to select headers and then to select an object by the same criterion. Selection of at least one component of the object is enough for selection of the whole object (not including nested MIME objects for a composite MIME object).



Note that it is required to **consider text encoding** for **all modification rules** that use a header or a part of a message.

Rules of working with header values in arbitrary encoding are described in [Work with String Values](#).



To select elements that satisfy several criteria, use logical operators for union or intersection of sequential selections. Specify several selection arguments in the `select` operator and join them with the following logical operators:

- **and** – leave only those elements in the selection that satisfy the specified criterion.
- **nand** – leave only those elements in the selection that do NOT satisfy the specified criterion.
- **or** – add elements that satisfy the specified criterion to the previous selection.
- **nor** – add elements that do NOT satisfy the specified criterion to the previous selection.



Note that join and intersection operators are used only in selections that contain MIME objects but are not used in selections that contain actual elements, as the operators cannot be applied to the `select mime.<SEGMENT>` syntax.

Example:

Select elements written in html and containing the "<script" word:

```
select mime(headers) Content-type html and mime(body) "\<script"
```

Actually, these are two sequential selections. The first selection contains all MIME objects that contain "html" word in the `Content-type` header. The second selection leaves only those elements that contain "<script" in any case.

Example:

```
select mime(headers) Content-type html nand mime(body) "\<script"
```

According to the first criterion, all MIME objects that contain "html" word in the `Content-type` header are selected. According to the second criterion, all MIME objects that contain "<script" in any case are excluded.

Example:

```
select mime(headers) Content-type html or mime(body) "\<script"
```

According to the first criterion, all MIME objects that contain "html" word in the `Content-type` header are selected. According to the second criterion, all MIME objects that contain "<script" in any case are added.

Note the following typical errors in using selection operators:

- If you specify several consecutive `select` operators, the successive `select` replaces the result of the previous `select`.

Example:

```
select mime(headers) Content-type html, select mime(body) "\<script"
```

```
select mime(headers) Content-type html and select mime(body) "\<script"
```

The received selection results correspond only to the second criterion, thus, only those mail fragments (MIME objects) are selected that contain "<script" in any case.

- If neither logical operators are used, nor `select` command is specified before the successive argument, this argument is ignored and selection results are not changed.

Пример:

```
select mime(headers) Content-type html mime(body) "\<script"
```

The received selection results correspond only to the first criterion, thus, only those mail fragments (MIME objects) are selected that contain "html" in the `Content-type` header.



2. Action operators

Actions are always applied to the result set of the `select` operator. These actions are divided into the following three types:

- Actions applied to the whole message (reject, redirect, generate MailD notifications and others);
- Actions applied to the selected elements of the message (deletion, signing, replacement or modification of text and others);
- Actions changing the message score (for anti-spam).



If a set of selected elements is empty, actions specified in the rule are not applied (are ignored).

2.1. Actions applied to the whole message

The following action operators applied to a whole message are available:

- **pass**, **accept** – pass the message. If **Dr.Web Modifier** is processing a message with the use of global rules and receives this command, the processing stops. Otherwise, if **Dr.Web Modifier** receives the command when using local rules, the plug-in starts using global rules for the processing;
- **reject** – reject the message and notify the sender;
- **discard** – reject the message without notifying the sender;
- **notify** <report_name> – notify the administrator; at that, message processing does not stop. After this command, specify the [notification template](#) name used to generate the notification, otherwise errors will occur during message processing. Templates are located in the directory specified in the `TemplatesBaseDir` parameter value in the [Notifier] section of the [Dr.Web MailD configuration file](#).

Example:

```
GlobalRules = select message, notify rule
```

In this case, only MailD notification is sent. As any action operator must be preceded by a selection operator, use `select message` operator which selects a whole message and returns non-empty result. The required `admin` prefix and `.msg` extension are [substituted automatically](#) by **Dr.Web Notifier**.

- **tempfail** – notify the sender on server failure;
- **redirect** – redirect the message to the specified address;
- **quarantine** – move the message to **Quarantine**.
- **stop** – stop processing modification rules. Action of the last performed command (`pass`, `accept`, `reject` or another) is applied to the message. The `accept` command is the same as `pass+stop` combination except for the fact that `stop` command stops full processing and `accept` stops only processing with local modification rules. For global modification rules, `accept` is equal to `pass`.

The `reject`, `discard` and `tempfail` commands are "resolving", that is, they stop message processing regardless of other commands specified in the modification rule.

2.2. Actions applied to the selected mail elements

These actions are applied only to content of selected elements unless otherwise specified.

- **replace** <replacement expression> <regular expression to be replaced>, **replace_all** <new text> – These actions replace text in the selected elements. The old text must match the <regular expression to be replaced> and the new text is specified in <replacement expression>.

**Example:**

Search and renaming of executable files in attachments (content of Content-disposition headers is replaced):

```
select mime.headers Content-disposition "filename=.*\\.exe",\  
or mime.headers Content-type "name=.*\\.exe",\  
replace "\\\\.ex_" "\\\\.exe", pass
```

These commands cannot be applied to composite parts of a message. That is, the following modification rule has no effect to a message that contains a composite MIME object with two nested objects (as **Container** shown in the figure above):

```
select message, replace_all "text"
```

because composite objects do not contain any data; they only serve as containers for other objects.

For **replace** and **replace_all** commands in the `<replacement expression>` and `<new_text>` arguments you can use function calls as `#{func_name}`. Argument for these functions is the current replacement expression. The following functions are implemented:

- o **urlencode** – encoding the argument to a string that can be used as URL;
- o **self** – return the expression unchanged.

Example:

```
select mime.headers "Subject" "^.*$", replace_all "[SPAM] #{self}"
```

At the start of **Subject** heading, the "[SPAM]" string is inserted. For example, "This is Subj" header is replaced with "[SPAM] This is Subj".

Example:

```
select mime.body ".*", replace "http://check-url.com?url=#{urlencode}"  
"http://\\S+"
```

In the message body, text that corresponds the specified template, for example: "Visit `http://vasya.com?id=3`", is replaced with "Visit `http://check-url.com?url=http%3A%2F%2Fvasya%2Ecom%3Fid%3D3`".

- **remove** – this command instructs to delete selected objects of any kind except for a root MIME object "message" (that is, this command cannot delete a whole message!).

Example:

In the following rules, **remove** command cannot be used:

```
GlobalRules = select mime (body) "text", remove, pass  
GlobalRules = select mime (body) "script", remove, pass
```

- **prepend_text**, **append_text**, **prepend_html**, **append_html** – These commands add fragment of `plain-text` or `html` format to the selected MIME objects.

The command has an optional argument - `[[7b:]encoding]`. In this argument, `encoding` means name of the added text encoding, and "7b:" prefix indicates use of 7 bit encoding - context transfer encoding (CTE) 7bit. If the "7b:" prefix is not specified, the CTE 8bit encoding is used. If `encoding` is not specified, the encoding set in the **Encoding** parameter in the plug-in [configuration file](#) is used.

Example:

```
select message, append_html "<h1>checked by anti-spam</h1>"
```

Apart from that, you can use **Dr.Web Modifier** language file as a source of data in certain



encoding. To use lines from the `lng`-file, specify them in the `$n` format, where `n` is the ordinal number of the string in the `lng`-file. For details on [language files usage](#), see description of the [Notifier] [section](#).

Example:

Let `lng`-file have the following line:

```
...
782 = "text line"
...
```

then `append_text $782` expression is equivalent to the expression `append_text "text line"`.



Note that `prepend_text`, `append_text`, `prepend_html`, `append_html` commands insert into the selected objects not only the text but the MIME object with text content (to the start or end of the selected objects respectively) and the modified objects become composite. Thus, adding of text information always results in reset of the selected objects list. To continue message processing after the text is inserted, repeat the `select` operation.

- **addheader** – Adding of headers into selected MIME objects.

Example:

```
select message, addheader "foo:bar"
```

This command adds a header with the `foo` name and `bar` value to the message. Name and value of the header are separated by a colon ("`:`").

It is required to consider text encoding both for text search and text insertion. Text is always inserted in the same encoding as one of the configuration file (except for `prepend_text`, `append_text`, `prepend_html` и `append_html` commands using for insertion either encoding specified in the command or encoding set in the `Encoding` parameter of the plug-in configuration file). To use another encoding, [explicit encoding of string values](#) is required.

2.3. Actions changing message scores

You can check or change score of a message. A message has a `score` assigned to it and originally set to 0. When processing the message, plug-ins can change this score. With `add_score` and `set_score` commands, this score can be changed, increased or decreased (to decrease a message score, specify a negative number in the `add_score` command).

Example:

```
set_score 10
```

Sets a message score to 10.

Example:

```
add_score 11
```

Increases a message score by 11.

The `score` argument can be 32 bit number in the range `-2 b` to `+2 b`. You should also consider that `score` can get too large value that exceeds the limit and thus causes incorrect operation of plug-ins that are processing the message. Thus, it is recommended not to set an unreasonably large value to a score (for example, do not specify `2 000 000 000` as a parameter of an `add_score` action)

2.4. Applicability of actions to different message objects

Table 1 shows effects of actions depending on the selected objects they are applied to.

**Table 1. Actions applied to selected elements of different types**

	remove	replace	replace all	append text	prepend text	append html	prepend html	add header	add score	set score	accept	discard	reject	tempfail	notify	redirect	quarantine
mime.headers	+	+	+	-	-	-	-	-	+	+	+	+	+	+	+	+	+
mime.prologue	+	+	+	-	-	-	-	-	+	+	+	+	+	+	+	+	+
mime.epilogue	+	+	+	-	-	-	-	-	+	+	+	+	+	+	+	+	+
mime.body	+	+	+	-	-	-	-	-	+	+	+	+	+	+	+	+	+
mime(headers)	+	*	*	+	+	+	+	+	+	+	+	+	+	+	+	+	+
mime(prologue)	+	*	*	+	+	+	+	+	+	+	+	+	+	+	+	+	+
mime(epilogue)	+	*	*	+	+	+	+	+	+	+	+	+	+	+	+	+	+
mime(body)	-	*	*	+	+	+	+	+	+	+	+	+	+	+	+	+	+
sender	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+
recipient	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+
message	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+

In the table, the following conventions are used:

- * – the same as for `mime.body`;
- + – applicable;
- – ignored.

Note that `prepend_text`, `prepend_html`, `append_text`, `append_html` and `addheader` actions can be applied only if a composite object (instead of a simple object, such as a header) is selected with the `select` operator, as these commands always insert into the selected object an additional MIME object which cannot be a header value.

3. Conditional validators

Conditional validators are used to change the order of actions applied to the message depending whether the condition meets the validation rule or not. There are the following main types of conditions:

1. Branching depending on selection results:

```
if [not] found, <action or list of actions>, \
[else, <action or list of actions>,) endif
```

The `if found` condition is true if the result set of the previous `select` operator is not empty. The `if not found` condition is true in the contrary case. When the condition is true, actions specified afterwards are applied until either the end of branching (`endif` operator) or an alternative branch operator is found. The alternative branch `else, <action or list of actions>` can be absent. In this case, when the condition is false, execution continues after the `endif` statement.

In the `<action or list of actions>` part, a new selection can be formed with the `select` operator and its condition is validated (`if` or `goto`). The new selection substitutes the previous one, which was used for condition validation). For example:

```
select <A>, if found, select <B>, reject, endif
```

In this case, set of elements `<A>` is selected. If the selection is empty, execution goes to `endif` and



stops (no action is performed to the message). Otherwise, new selection is formed from the message, specified as (at that, it substitutes the previous selection <A> for subsequent operators). If the selected set is not empty, the message is rejected. Otherwise, no action is performed to it (**reject** action is ignored for an empty selection).

2. Branching depending on a message score:

```
if score <op_value>, <action or list of actions>, \  
[else, <action or list of actions>] endif
```

The **if score** condition is true if the current message score corresponds to the specified expression. The **if score** command is equal to the **if found** command except for the fact that **if score** checks only the score and does not check presence of selected elements, that is this command ignores results of the previous selection.

The <op_value> argument must be specified on a single line without space characters, that is '<100', but not '< 100'. It must consist of a symbol of comparison and an integer value. For **if score**, the following comparisons are available:

- **if score** <N – if score is less than N;
- **if score** >N – if score is bigger than N;
- **if score** =N – if score equals to N.

N argument can be a 32-bit integer value in the range -2 b to +2 b. You should also consider that **score** can get too large value that exceeds the limit and thus causes incorrect operation of plug-ins processing the message. Thus, it is recommended not to set an unreasonably large value to **score** (for example, do not set 2 000 000 000).

3. Jump statements:

- **goto** N – unconditionally jump forward N commands in the action list
- **goto** (y) N – if the condition is true, jump forward N commands in the action list if the set of elements is not empty
- **goto** (n) N – if the condition is true, jump forward N commands in the action list, if the set of elements is not empty.

The argument denotes a number of commands to be skipped. To specify the argument, use a positive integer.

Examples:

```
GlobalRules = select mime.headers Subject "word1|word2|wordN", \  
if found, notify rule, quarantine, reject, endif
```

When one of "word1", "word2" or "wordN" words is found in the message header, the message is copied to **Quarantine**, after that a MailD notification is send to the administrator and the message is rejected.

```
GlobalRules = select mime.headers Subject "word1|word2|wordN", \  
if found, reject, notify rule, quarantine, endif
```

In this example, **notify** and **quarantine** commands are not executed as after the **reject** command the message is rejected and its processing stops (**reject** command is "determining").

```
GlobalRules = select mime(header) Content-type "executable", \  
goto(n) 1, reject
```

This command set allows rejecting messages with executable files attached.



[Filter] [section](#) of the **Dr.Web MailD** configuration file.

Example:

```
AfterQueueFilters = modifier
```

Setting Modifier Plug-In

All main parameters that regulate plug-in operation are set in the `%etc_dir/plugin_modifier.conf` configuration file. Description of the configuration file structure and parameter types can be found in [Configuration Files](#). Parameters are described in the order they appear in the main configuration file.

In the [Modifier] section, general settings for **Dr.Web Modifier** plug-in are specified:

[Modifier] section

<pre>GlobalRules = {list of rules}</pre>	<p>List of global modification rules used by the plug-in for message processing.</p> <p>Example:</p> <p>The rule below adds a note in the html format to the message:</p> <pre>GlobalRules = select message, append_html "checked!"</pre> <p>The rule below deletes messages from the specified users:</p> <pre>GlobalRules = select mime(headers) From "weirdohacker@server.net", if found, reject, endif</pre> <p><u>Default value:</u> GlobalRules =</p>
<pre>Encoding = {text}</pre>	<p>R Encoding that the plug-in uses for the text inserted with <code>append_text</code> and <code>prepend_text</code> commands directly from modification rules.</p> <p><u>Default value:</u> Encoding = koi8-r</p>

When a message is blocked (`reject`) by the plug-in in the [synchronous mode](#), **Dr.Web MailD** response to a client contains SMTP code (55* or 250, depending on the `ReturnReject` parameter value in the [Receiver] [section](#)) and a text message which content is determined by values of the parameters described below. Their values must be enclosed in quotation marks.

<pre>UseCustomReply = {logical}</pre>	<p>R Use custom messages as an SMTP reply specified in the <code>ReplyRuleFilter</code> parameter when messages are rejected by Dr.Web Modifier.</p> <p><u>Default value:</u> UseCustomReply =</p>
<pre>ReplyRuleFilter = {text}</pre>	<p>R Custom message used as an SMTP reply when the message is rejected by Dr.Web Modifier.</p> <p><u>Default value:</u> ReplyRuleFilter =</p>

If `UseCustomReply = No` or the corresponding string is not specified, the following standard message outputs: "The message has been rejected by the Dr.Web MailD".



Examples

Examples of setting rules for **Dr.Web Modifier** (by the example of global rules):

- 1. Select elements what satisfy two certain conditions. If these elements are present, remove the message. Otherwise, find all executable files attached to a message and delete them. After that, add the specified text to the end of the message:**

```
GlobalRules = select mime(headers) Content-type "text" \  
and mime(body) "typical spam", goto(n) 1, discard, \  
select mime(headers) Content-disposition ".exe", \  
remove, select message, append_text "checked!"
```

- 2. Delete messages received from the specified users:**

```
GlobalRules = select mime(headers) \  
From "weirdohacker@server.net", if found, \  
reject, endif
```

- 3. Redirect messages received from the specified users:**

```
GlobalRules = select mime.headers \  
To "someaddress@my-net.com", \  
redirect "anotheraddress@my-net.com"
```

In this case, original message is delivered to someaddress@my-net.com user and copy of this message is delivered to anotheraddress@my-net.com user. If you do not need to deliver the message to the original recipient, use the rule described below.

- 4. Select messages by certain conditions, redirect the selectes messages to the specified address and delete original messages in order not to deliver them to their original recipients:**

```
GlobalRules = select mime.headers Subject "Help", \  
if found, select mime.headers To "someaddress@my-net.com", \  
if found, redirect "anotheraddress@my-net.com", \  
discard, endif, stop, endif
```

- 5. Redirect messages received on the corporate mail box depending on the subject:**

- a) Messages with subject satisfying to the expression "support|problem" are redirected to technical support;
- b) Messages with subject satisfying to the expression "price|sale|buy" are redirected to sales department;
- c) All other messages are redirected to the main mail box for incoming messages:

```
GlobalRules = \  
select mime.headers Subject "support|problem", \  
if found, select mime.headers To "@company.com", \  
if found, redirect "support@company.com", \  
endif, pass, endif, \  
select mime.headers Subject "price|sale|buy", \  
if found, select mime.headers To "@company.com", \  
if found, redirect "sell@company.com", \  
endif, pass, endif, \  
select mime.headers To "@company.com", \  
redirect "inbox@company.com", pass
```



6. Find and rename all attached executables:

```
GlobalRules = select mime.headers \  
Content-disposition "filename=.*\\.exe", \  
or mime.headers Content-type "name=.*\\.exe", \  
replace "\\ex_" "\\exe", pass
```

Work with String Values

Note that it is required to **consider encoding** of the search text in **all rules** and plug-in **parameters** that use a message header or body.

Values that are specified directly and not in Latin can match only if the search text encoding is the same as the encoding used in the configuration file. If it is required to use values in another encoding, insert a string transformed into the required encoding and coded using a transport code, for example **base64**. The final string value must be of the following form:

```
=?<source_enc>?<B|Q>?<coded_text>?="
```

where:

- <source_enc> – source text encoding (for example, UTF-8, CP1251);
- <B|Q> – indicator of the used transport encoding (B – base64, Q – quoted-printable);
- <coded_text> – encoded string value.

Note that if the result value must be treated as a regular expression, escape regular expression symbols (for example, ? inquiry character) with double backslash '\\'.

Example:

Suppose it is required to search messages that contain the word "тест" entered in the Russian language using the Cyrillic script in CP1251 encoding. After the text is translated into **base64** transport encoding, the string is as follows: 8uXx8go=.

To get strings in the required encoding, use special tools, for example, **iconv** and **base64**. The following example translates the "тест" character string from UTF-8 encoding to CP1251, and then presents the string with the use of **base64**:

```
echo "тест" | iconv -f utf-8 -t cp1251 | base64
```

Example of usage in plug-ins:

1. To enable **Dr.Web Modifier** to select a header that includes "тест" in CP1251 encoding, specify the following condition in the modification rule:

```
select mime.headers Subject "=\\?cp1251\\?B\\?8uXx8go=\\?="
```

2. To enable **Vaderetro** to add the word "тест" in CP1251 encoding to message headers, specify the following:

```
SubjectPrefix = "?cp1251?B?8uXx8go=?="
```

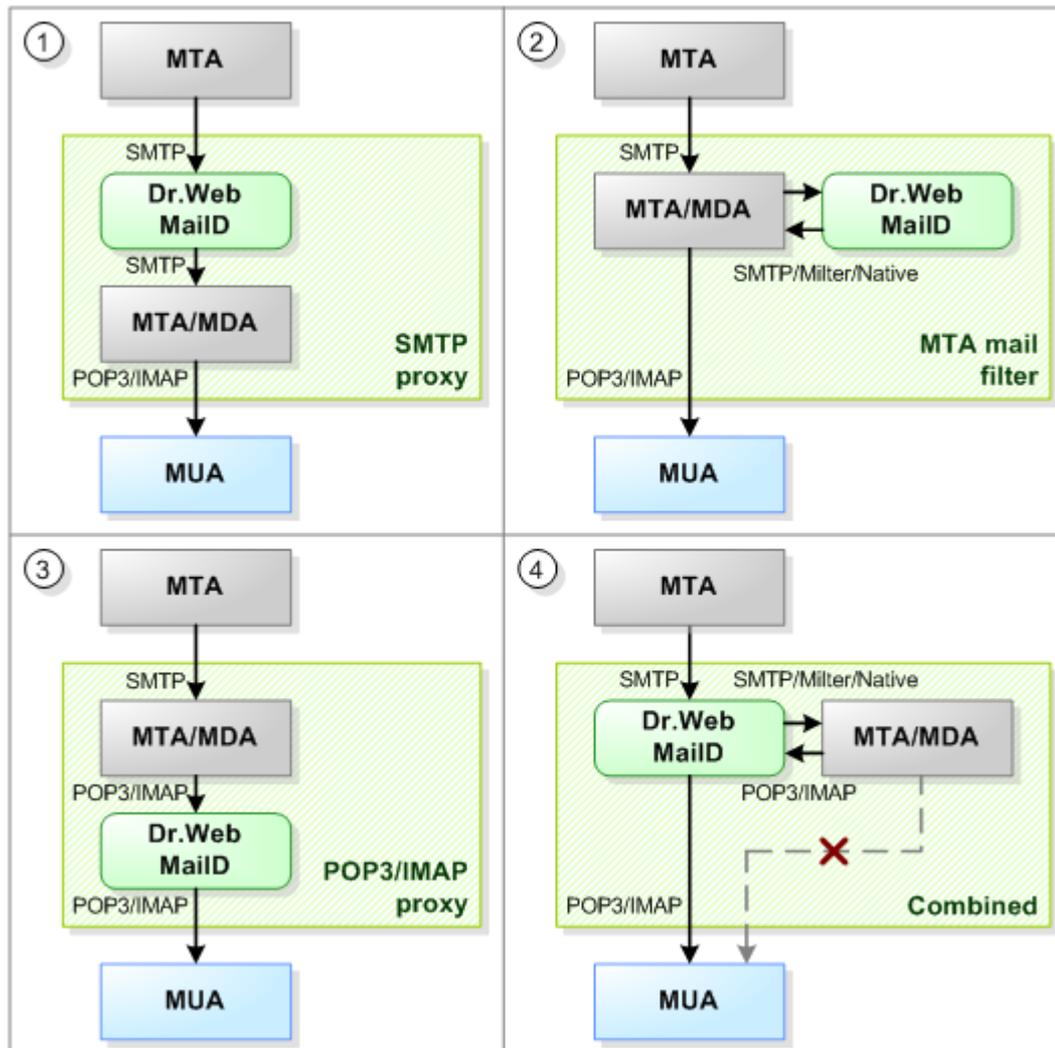
3. To enable **Dr.Web HeadersFilter** to reject messages, containing in subject the word "тест" in CP1251 encoding, specify the following:

```
RejectCondition = Subject = "=\\?cp1251\\?B\\?8uXx8go=\\?="
```

Integration with Mail Transfer Systems

This chapter provides you with information on features of **Dr.Web for UNIX mail servers** integration with different mail transfer systems.

Three methods of **Dr.Web for UNIX mail servers** integration are available. They are shown in the following picture (the fourth integration method, presented in the picture, is combined).



Picture 20. Methods to integrate Dr.Web for UNIX mail servers with mail systems

Note that all methods to integrate **Dr.Web for UNIX mail servers** directly with mail systems use only components mail processing – **Dr.Web MailD**.

1. **SMTP/LMTP proxy** integration. Basic integration method, which is universal and applicable to all cases. It is suitable for integration with any MTA, as the method uses only standard mail protocols SMTP/LMTP. In this case, **Dr.Web MailD** is a proxy between an external MTA, which sends mail correspondence to the server, and an internal MTA/MDA, which is responsible for further storage of checked email messages and interaction with recipients (MUA) or transmitting messages to other mail systems. When required, you can also use this mode in order to organize checking of messages in [SMTP Callback mode](#) (note that for this the additional settings are required).

Note that this integration method does not require a protected MTA to run on the same server where **Dr.Web for UNIX mail servers** operates. For details on how to configure this integration method, refer to [Working in SMTP/LMTP Proxy Mode](#).



2. **MTA Mail filter** integration. When this integration method is used, mail system performs communication with external MTAs that send mail correspondence to the server. Moreover, mail system is responsible for storage of checked email messages and communicates with message recipients (MUA). At that, **Dr.Web MailD** is used only as an external application-filter that checks received messages transmitted by the mail system. **Dr.Web MailD** returns check results and they determine further actions to be applied to the message.

For interaction between **Dr.Web MailD** and mail system in the filter mode, both standard protocols (for example, `Milter` and `SMTP`) and native protocols specific to the certain mail system can be used. For that, **Dr.Web MailD** includes [special interaction plug-ins](#), implemented for connection to some mail systems in the filter mode. In this mode, **Dr.Web MailD** can be integrated with the following mail systems:

- **CommuniGate Pro** (see [Description of integration configuration](#));
- **Sendmail** (see [Description of integration configuration](#));
- **Postfix** (see [Description of integration configuration](#));
- **Exim** (see [Description of integration configuration](#));
- **Qmail** (see [Description of integration configuration](#));
- **Courier** (see [Description of integration configuration](#));
- **Zmailer** (see [Description of integration configuration](#));

When it is possible to integrate **Dr.Web MailD** with a mail system, the **MTA Mail filter** mode is preferred to the **SMTP proxy** universal mode, as the filter mode requires less load on calculating server capacity: in this case, **Dr.Web for UNIX mail servers** performs only anti-virus and anti-spam functions and is not responsible for receiving and sending mail correspondence.

The **MTA Mail filter** integration method assumes that the protected MTA is running on the same server where **Dr.Web for UNIX mail servers** operates.

3. **POP3/IMAP proxy** integration. In this mode, **Dr.Web for UNIX mail servers** is used for message check upon transferring the message to the MUA of the end recipient via `IMAP` or `POP3` mail protocol (and not at the moment when the message is received). This integration solution can be implemented only if the mail system protected with **Dr.Web for UNIX mail servers** is not a proxy but is finite, meaning that the system serves requests from the end MUA.

In this case, **Dr.Web for UNIX mail servers** is embedded as a proxy between MUA and MDA and transfers messages returned to MDA at the MUA request to check them by **Dr.Web MailD**. This method does not require MDA and MUA to run on the same server as **Dr.Web for UNIX mail servers** does. For details on how to configure this integration method, refer to [Working with POP3/IMAP Mail Clients](#).

4. **Combined** integration. This mode is a combination of either the first or the second mode (any of them) with the third mode. Thus, it is possible, although not always reasonable, to implement double message check - when a message is received on the protected MTA/MDA and when the message is transferred from MDA to MUA. This integration method can be useful if some of the messages are sent through MTA further (and they must be checked on the SMTP level), but for the rest of the messages this MTA is the end MDA and they can be checked by request from the end MUA upon receipt.

Note that in this case fine tuning of the **Dr.Web MailD** operation logic is required, including specification of [message processing Rules](#) and, probably, rules for routing the outgoing mail (set in the `Router` parameter) in [Sender settings](#).

To simplify the integration process, **Dr.Web for UNIX mail servers** includes installation packages and configuration scripts for different mail systems.

The `configure_mta.sh` script is responsible for setting up interaction between **Dr.Web for UNIX mail servers** and the currently used mail system. After startup, the script checks whether the required mail system is installed. If it appears to be missing, the script finishes its operation. If the required mail



system is installed, the script asks the user several questions on essential settings for basic setup. Setup can be performed manually as well (for details, refer to the corresponding chapters of this manual).

The `configure_mta.sh` script configures MTA as follows:

- [Connection using special transport](#) is performed for **Exim**
- [After-Queue Mode](#) configuration is performed for **Postfix**
- **Zmailer** is configured to be used in [context filter mode at the stage of SMTP-session](#)
- [Proxy scheme](#) is performed for **Qmail**.

Thus, for example, to set **Postfix** to operate using `Milter` protocol, configure MTA according to the steps [described in the corresponding section](#) instead of running the `configure_mta.sh` script.

Working in SMTP Proxy Mode

Dr.Web MailD can operate as a proxy server for SMTP/LMTP mail protocols, which allows to use it with a great number of mail systems. In this mode, `drweb-receiver` [module](#) operates as an SMTP/LMTP server (i.e., **Receiver** component) and `drweb-sender` module operates as an SMTP/LMTP client (i.e., **Sender** component).



As `drweb-sender` module can transfer messages directly via the local mail system, the module is used in almost any solutions of integration with MTA.

Module `drweb-receiver` includes a high-performance SMTP server, implemented using modern multiplexors (`epoll`, `kevent` and `/dev/poll`). This SMTP server is multithreaded. It supports several connections per each thread, IPv6 protocol and the following SMTP extensions:

- PIPELINING ([RFC 2920](#))
- 8BITMIME ([RFC 1652](#))
- ENHANCEDSTATUSCODES ([RFC 3463](#))
- SIZE ([RFC 1870](#))
- AUTH ([RFC 4954](#))

When **Dr.Web MailD** operates in the **SMTP/LMTP proxy** mode, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of the **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-receiver`
- `drweb-maild`

All settings for `drweb-receiver` and `drweb-sender` modules are collected in the `[Receiver]` and `[Sender]` sections of the **Dr.Web MailD** configuration file and described in [\[Receiver\]](#) and [\[Sender\]](#) sections of the current Manual.

Also it is additionally necessary to make, that the **Dr.Web Monitor component** has been launched with the `root` privileges (to provide this, specify `root` value for **User** and **Group** parameters in the `[Monitor]` [section](#) of `monitor.conf` configuration file).

When mail is received directly from the Internet, `drweb-receiver` uses implemented anti-spam technologies that make mail filtering easier and more efficient and allow filtration on the SMTP session stage:

- [SMTP session restriction control](#) technology.



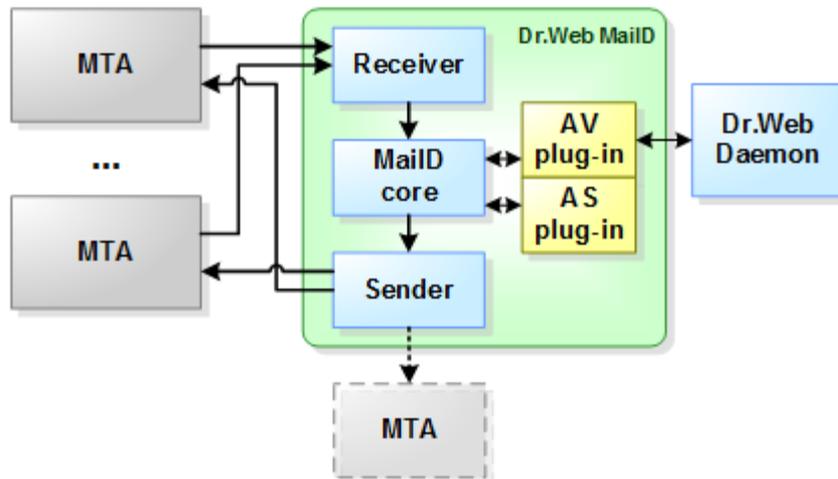
- Technologies to estimate reputation of the connection (sender) - [Reputation IP Filter](#) and [Unified Score](#).

See also recommendations on using [synchronous and asynchronous modes](#) of message processing.

SMTP Callback Mode

Dr.Web MailD can operate as both a proxy server for SMTP/LMTP protocol and as a service for checking emails in the Callback mode. In this mode, MTA, which receives an email message, transmits it to **Dr.Web MailD** for check via the SMTP/LMTP protocol. Having checked the message, **Dr.Web MailD** returns it to the same server, which transmitted the message, via the SMTP/LMTP protocol.

The scheme of the Callback mode implementation is shown in the picture below.



Picture 21. Integration between Dr.Web MailD and mail systems in the Callback mode

The set of running [components](#) and main settings are similar to those of the **SMTP/LMTP proxy integration mode**.

Special Settings of Callback Mode

The feature of the Callback mode settings is that **Sender** must return the email message to the same host which sent this message to **Receiver** (but to another port). For that purpose, add special [Rules](#) that modify the procedure of sending messages. The rules are specified in the [Rules] [section](#) of the main **Dr.Web MailD configuration file**. When specifying the Rules, use the [SenderAddress parameter](#) to set the address to which checked messages are returned.

Specify the Rules as follows:

```
<CONDITION> cont SenderAddress = inet:<port-num>@CLIENT-IP
```

where <CONDITION> - condition when the Rule is executed (in the most trivial case, specifying `true` is sufficient), and <port-num> - number of the port where MTA is waiting for a message (for example, 10025). CLIENT-IP - special macro that is substituted with the address which **Receiver** stores as the address from which the message was received.

Note that all actions applied to messages by **Receiver** and [plug-ins](#) must be configured so as to avoid denial to receive messages for check (do not use the following [actions](#): `reject`, `discard`, `tempfail`). Alternatively, you must adjust these MTA so that they can correctly handle the cases, when the callback filter rejects or discards the message.

Moreover, if **Dr.Web MailD** is configured to enable generation of DSN (for example, when the operation mode is [asynchronous](#) and the [SkipDSNOnBlock parameter](#) value is set to `No`) and



[notifications](#), the <CONDITION> part of the Rule must contain the validation condition that does not allow service messages to be sent to the address where MTA is waiting for the checked messages. For that purpose, specify validation conditions which contain the **from (sender)** parameter. All DSN are always sent with the empty `From` field and the `From` field of notifications contains the address specified in the `FilterMail` parameter value (`[Notifier]` [section](#)).

Example:

```
!(from:"" || from:"root@localhost") cont SenderAddress = inet:10025@CLIENT-IP
```

This Rule sends an outgoing message to the port 10025 on the sender's host only if the `From` field is empty or does not contain the `root@localhost` address.

The `SenderAddress` parameter, used in rules, overrides the `Address` parameter value (specified in the `[Sender]` [section](#)) dynamically. Thus, if the Rule is not executed, the message will be sent to the MTA specified in the **Sender** settings. It is recommended to specify an active MTA in the settings to rapidly receive messages notifying on problems that occurred upon email processing (in the picture above, this MTA is indicated with a dashed line).

Working with POP3/IMAP Mail Clients

Dr.Web for UNIX mail servers can be used for checking messages not when mail system receives them but at the moment they are transferred via `IMAP` and `POP3` to the MUA of the end receiver. This integration solution can be implemented only when the mail system protected with **Dr.Web for UNIX mail servers** is not a proxy, but is finite, meaning that the system serves requests from the end MUA.

To implement the solution, the following two proxy components are included into **Dr.Web for UNIX mail servers**:

- **POP3 filter** – used for intercepting messages of `POP3` the protocol during communication between MUA and MDA. Implemented as `drweb-pop3` module;
- **IMAP filter** – used for intercepting messages of `IMAP` protocol during communication between MUA and MDA. Implemented as `drweb-imap` module.

The following picture illustrates **Dr.Web MailD** connection diagram when working with mail clients.

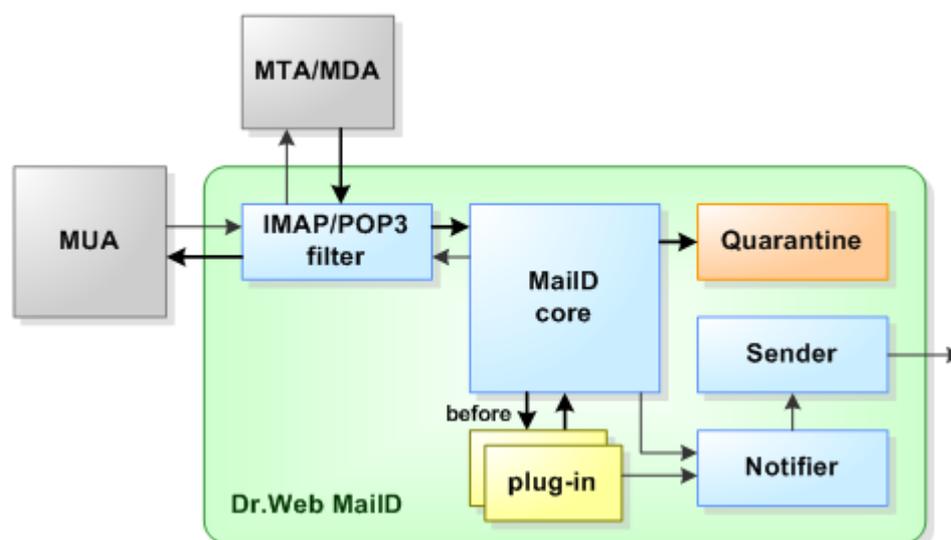


Рис. 22. Work with mail clients



General operation principles:

1. Filter of the user protocol (**POP3 filter** or **IMAP filter**, depending on the used protocol) is configured as a proxy to receive messages transmitted from MUA to MDA through the corresponding protocol.
2. Messages received from MUA are transferred to the target MDA (it can be local or remote in relation to **Dr.Web MailD**).
3. After a message is transmitted from MDA to MUA, it passes through the filter component which sends the message for check to **MailD core** (via the interface used by **Receiver**).
4. **MailD core checks the message** (using **Processing rules** and configured **plug-ins**).
5. If **MailD core** responds with the positive check result, the message is transmitted further to MUA. Otherwise, instead of the requested message, MUA receives a report on the detected threat. This report is generated by **Notifier**.
6. If **MailD core** is enabled to send reports on detected threats, the corresponding reports generated by **Notifier** are dispatched by **Sender** (it sends the reports for final delivery to MTA that is specified in **Sender** settings).

Note the following **restrictions** when **Dr.Web MailD** is used for checking messages via client protocols:

- All used plug-ins must be assigned to the **BeforeQueueFilters** **queue**, that is, message check in **asynchronous mode**, when messages are saved to the storage, is not allowed (due to aspects of POP3 and IMAP operation).
- `redirect` **action** must not be used in Rules and plug-in settings, as a message transmitted from MDA to the user's MUA cannot be redirected to another address.

Filter component characteristics:

1. IMAP filter component

Supports interaction with IMAP servers (including the cache function). This component is a proxy server between **MailD core** (`drweb-maild`) and IMAP server (MDA). The component filters messages which the server sends to the user. MDA IMAP server can run on the local computer, as well as on a remote computer.

Functions of the component are performed by the `drweb-imap` **module**. Its settings are specified in the `[IMAP]` **section** of the main **Dr.Web MailD configuration file**.

The **IMAP filter** component caches main message headers to speed up access to them. Theoretically, it is possible to run out of available memory and slow down filter operation by filtering large number of messages that are formed in a special way and contain a lot of headers.

To prevent this situation, **IMAP filter** has `MaxCachedHeadersPerMail` setting that controls maximum total size of cached headers. Note that if this value is too small, names and types of MIME attachments can display incorrectly on users' computers.

The filter is disabled by default. To enable it, uncomment the following string in the **mmc file** of **Dr.Web Moritor** (`maild_<MTA>.mmc`):

```
drweb-imap local:/var/drweb/ipc/.agent 15 30 MAIL drweb:drweb
```

2. POP3 filter component



Supports operation with POP3 servers. The component is a proxy server between **MailD core** (`drweb-maild`) and POP3 server (MDA). The component filters messages which server sends to the user. MDA POP3 server can run on the local computer, as well as on a remote computer.

Functions of the component are performed by `drweb-pop3` [module](#). Its settings are specified in the [POP3] [section](#) of the main **Dr.Web MailD configuration file**.

Every time a connection is established, **POP3 filter** retrieves the user name from the `USER username` command and saves the name during the session. If authentication is successful, the filter performs transmission of the messages from the server to the client. At that, all commands and data are transmitted literally except for a server response to the `RETR` command (message retrieval).

Response from MDA to this command is transmitted to **MailD core** for analysis and then MUA receives the processed response.

The filter is disabled by default. To enable it, uncomment the following string in the [mmc file](#) of **Dr.Web Moritor** (`maild_<MTA>.mmc`):

```
drweb-pop3 local:/var/drweb/ipc/.agent 15 30 MAIL drweb:drweb
```

When **Dr.Web MailD** is operating in the **POP3/IMAP proxy** mode, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of the **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`
- `drweb-pop3` or `drweb-imap` (depending on the intercepting user protocol)



Note that a part of the `<MTA>` file name depends on the name of the MTA associated with **Dr.Web for UNIX mail servers**

Integration with CommuniGate Pro

Configuring CommuniGate Pro

To enable **CommuniGate Pro** (hereinafter **CGP**) to send and receive messages from **Dr.Web MailD**, do the following:

1. Connect to **CGP** web interface (it must be connected as a plug-in in a remote administration tool **WebAdmin**);
2. Select **Settings**, then select **General** and open the **Helpers** menu;
3. In the **Content Filtering** section, add a new external filter and specify the following parameters (the new filter is added in the last, empty, line in the filter list):

```
Enabled (select in the drop-down list)
Enter the filter name in the text field: DrWeb Maild
Log Level: Problems
Program Path: %bin_dir/drweb-cgp-receiver
Time-out: 2 min (select in the drop-down list)
Auto-Restart: 15 sec (select in the drop-down list)
```

4. Check whether the privileges with which **CGP** is executed are sufficient for `drweb-cgp-receiver` startup;



5. Select `Settings`, then select `Queue` and open the `Rules` menu;
6. Create a new rule.

To create a new rule, do the following:

1. Enter the rule name (for example, `drweb-filter`) and click the **Add Rule** button;
2. Click **Edit** and specify the `ExternalFilter` value in the **Action** drop-down list;
3. In the **Parameter** field enter the filter name specified in the previous step (on the `Settings` -> `General` -> `Helpers` menu). In the given example, the filter name is `DrWeb Maild`.

It is recommended to add additional settings to the created rule:

1. To avoid repeated check of messages received from `GROUP`, `LIST`, or `RULES` (<http://www.communicate.com/CommuniGatePro/Transfer.html>), add the following setting to the rule:

```
"Submit Address", "not in", "GROUP*,LIST*,RULES*"
```

To set the rule, select the checked field type and comparison operator from the drop-down list and enter the checked value in the text field.

2. When a message is uploaded through `PIPE`, the `authenticated` flag can be lost. Thus, if some [plug-ins](#) are assigned to the `AfterQueueFilters` [queue](#), add the following setting to the rule:

```
Any Recipient not in alldomains@main.domain,all@*
```

where `main.domain` is the main domain of the **CGP** server.

For information on advanced settings (for example, enabling or disabling filtering for a certain user), refer to the documentation distributed with **CGP**.

Configuring Dr.Web MailD

When interacting with **CGP**, `drweb-cgp-sender` module of **Dr.Web MailD** performs functions of **Sender**. The module is started with privileges of the `mail` group to enable writing to the `cgp` directory. `drweb-cgp-receiver` module of **Dr.Web MailD** performs functions of **Receiver**. The module is started by **CGP** mail system with `root` privileges.

To assure proper operation of **Dr.Web MailD** in this configuration, explicitly specify the name of the user with whose privileges other **Dr.Web MailD** modules are started. This name can be set in the `ChownToUser` parameter from the `[CgpReceiver]` [settings section](#) in the **Dr.Web MailD** configuration file, or you can specify an empty parameter value and run the whole suite with `root` privileges.

Interaction between **Dr.Web MailD** and **CGP** mail system has the following features: it is performed locally (via the `PIPE` mechanism), **Dr.Web MailD** performs functions of a content filter and, thus, cannot modify message headers. Therefore, when **Dr.Web MailD** needs to change message headers, for example, to mark a message as spam (usually by adding the "[SPAM]" string to the message subject), the following method is used: **Dr.Web MailD** sends **CGP** a notification with request to reject the original message and simultaneously adds the modified message to the queue of incoming email. The message is checked by **Dr.Web MailD** for the second time and the following actions are applied to prevent looping:

- The `drweb-cgp-receiver` module skips all messages received via `PIPE` without check. As `drweb-cgp-sender` loads new messages to **CGP** via `PIPE`, the repeated check is prevented. However, this results in skipping all other messages queued by any program via `PIPE`.
- To avoid skipping such messages, it is recommended to add a special header to them. This is configured by `UseSecureHash` and `SecureHash` parameters in the `[CgpSender]` [section](#) of the **Dr.Web MailD** configuration file. If the `UseSecureHash` parameter value is set to `Yes`, such



header with the `X-DrWeb-Hash` name is added to a message when being assigned to the queue of **CGP** incoming email. The `SecureHash` parameter value determines the text of the header.

- In this case, messages received from the mail system are transmitted for delivery bypassing the check if they both were queued via PIPE and contain the `X-DrWeb-Hash` header with the value specified in the `SecureHash` parameter. The `drweb-cgp-receiver` module transmits the messages for final delivery after the value is cleared (by substituting characters with white spaces). Messages without such header are transmitted for check.

Note that **Dr.Web MailD** operating as a content filter cannot remove headers; so, if a message was repeatedly checked, its end users receive the message with an empty (filled with white spaces) `X-DrWeb-Hash` header, which does not influence display of the message content.



Note that this parameter is used by both **Sender** and **Receiver**, so after the parameter value is changed, sending a `SIGHUP` signal to **Dr.Web Monitor component** is not enough (that instructs only **Sender** to reread its configuration). To instruct **Receiver** to reread the changed value, restart **CGP**, as this mail system runs the component.

All settings that manage operation of **Dr.Web MailD** with **CGP** are collected in the `[CgpReceiver]` and `[CgpSender]` sections of the **Dr.Web MailD** configuration file and are described in the `[CgpReceiver]` and `[CgpSender]` sections respectively.

When **Dr.Web MailD** is working with **CGP**, the following `modules` must be running in the system (this is specified in `mmc file` of the **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-cgp-sender`
- `drweb-maild`

Also it is additionally necessary to make, that the **Dr.Web Monitor component** has been launched with the `root` privileges (to provide this, specify `root` value for `User` and `Group` parameters in the `[Monitor]` `section` of `monitor.conf` configuration file).

Operation Principles

Dr.Web MailD interacts with **CGP** in the following way:

1. **CGP** receives an email message.
2. **CGP** checks the settings and after that, if required, sends the message for check to `helper`, functions of which are performed by `drweb-cgp-receiver`.
3. Upon receipt of the message, `drweb-cgp-receiver` searches for `SecureHash` header:
 - if the header is found, `drweb-cgp-receiver` returns `OK` status and passes the message to **CGP** for further processing;
 - otherwise, the message is passed for check to `drweb-maild`;
4. `drweb-maild` applies plug-ins to the message and the plug-ins can modify it (for example, add headers);
 - if no virus is detected and the message was not changed, `OK` status is returned to **CGP**;
 - if during processing the message was changed, `DISCARD` status is returned to **CGP** and further message processing is performed by `drweb-maild`, as the `helper` protocol does not allow to return a modified message.



- The message is passed to **Sender** and, after adding of `SecureHash` header (if `UseSecureHash = yes`), the message is moved to the submit directory `/var/CommuniGate/Submitted/` that is periodically checked by **CGP**.



Value of the **SubmitDir** parameter of the **Dr.Web MailD** configuration file must be equal to `/var/CommuniGate/Submitted`. Otherwise, messages checked by **Dr.Web MailD** cannot reach their recipients.

- After the `/var/CommuniGate/Submitted/` directory is checked and the message is received, **CGP** goes to step 2:
 - if the settings are correct, the message is not checked again;
 - if the settings are inaccurate, the message is returned to **CGP** after check of the `SecureHash` header;
 - if the settings are incorrect, an infinite checking loop can occur.

Known Issues

In **Linux** operating systems after command line is altered and updated with the `Helpers` setting, the previous filter process remains in the `zombie` status until **CGP** restart.

Description:

When `drweb-cgp-receiver` is started, the following messages are displayed:

```
/usr/libexec/ld-elf.so.1: Shared object "libstdc++.so.6"
not found, required by "libboost_thread.so"
```

Solution:

The system cannot find necessary libraries located in the `%bin_dir/lib/` directory. It is required to copy `libstdc++.so.6` and `libgcc_s.so.1` libraries (or make symbolic links to them) from `%bin_dir/lib/` to the system library directory.

Integration with Sendmail

For interaction between **Dr.Web MailD** and **Sendmail** system, the latter must support `Milter` API. If the used **Sendmail** copy does not support the API, rebuild the system to add `Milter` API to the supported libraries. For details, refer to the corresponding documentation on the used **Sendmail** system.

Note: To check whether the **Sendmail** copy supports `Milter` API, use the following command:

```
# sendmail -bt -d0 < /dev/null
```

If the output text contains "milter" string, your copy of **Sendmail** supports `Milter` API.



Dr.Web MailD is fully compatible with **Sendmail** versions 8.12.3 and later. When working with earlier versions, some problems can occur (see [Known Issues](#)). Detailed integration instructions provided in this documentation are suitable for **Sendmail** version 8.14.0 or later.



Interaction between **Sendmail** MTA and **Dr.Web MailD** is performed via `Milter` API (`drweb-milter` module is used as **Receiver**) and is implemented as follows:

- Through the transport connection defined by `drweb-milter` transport address `__ADDRESS__`, **Sendmail** system receives internal commands from `Milter` API and a message itself. The message is sent in segments depending on the stage of the mail session (`helo`, `mail from:`, `rcpt to:`, etc.). Therefore, the message is saved by `drweb-milter` module to the temporary directory. Through `Milter` API, `drweb-milter` transmits instructions regarding the message to the **Sendmail** system.

`Milter` API is a multithreaded library – several mail sessions can be processed simultaneously. In the interaction scheme, given above, **Sendmail** system is a client and `drweb-milter` is a server, therefore, in the `sendmail.cf` configuration file `drweb-milter` address must be specified, and **Sendmail** system chooses the appropriate client address for this connection;

- Through another transport connection `drweb-milter` module sends commands to `drweb-maild` module and waits for its response.

In the scheme given above, `drweb-milter` module works as a proxy (or transformer) between the **Sendmail** system interface and `drweb-maild` module.

Note [features of operation](#) through `Milter` in synchronous and asynchronous modes.



Sendmail and `drweb-milter` can operate on different computers, but `drweb-milter` and `drweb-maild` modules must operate on the same computer.

Configuring Sendmail

To set up interaction between **Sendmail** and **Dr.Web MailD**, changes to `sendmail.mc` and `sendmail.cf` configuration files are required.

To avoid recompilation of the `sendmail.cf` configuration file, you can insert or add there the following lines (if the corresponding definitions are already present in the file):

For versions 8.14.0 and later:

```
#####
# Input mail filters
#####
O InputMailFilters=drweb-milter
O Milter.LogLevel=6
#####
# Xfilters
#####
Xdrweb-milter, S=__ADDRESS__,
F=T,T=C:1m;S:5m;R:5m;E:1h
```

To check locally sent messages (with `mail` or `sendmail` system call), all changes made to `sendmail.cf` configuration file must be copied to `submit.cf` and `submit.mc` files.

Please note that `submit.cf` and `submit.mc` files are read-only by default, so you must change access permissions (providing write access) before making any changes to these files. Moreover, you must add `nobodyreturn` value to the `O PrivacyOptions` parameter.

**Example:**

```
# privacy flags
O PrivacyOptions=goaway,noetrn,nobodyreturn
```

Or in {sendmail_src}/cf/cf/feature/msp.m4:

```
define(`confPRIVACY_FLAGS'
`goaway,noetrn,nobodyreturn,restrictqrun')
```

If the filter is not available, you can enable the following flags (F=):

- R - fail to deliver;
- T - delay delivery.

If neither F=R, nor F=T is specified, the message is passed without check.

You may also add the following lines to `sendmail.mc`:

For versions 8.14.0 and later:

```
INPUT_MAIL_FILTER(`drweb-milter', `S=__ADDRESS__,
F=T, T=C:1m;S:5m;R:5m;E:1h')
define(`confMILTER_LOG_LEVEL', `6')
```

Timeout must be set according to the values of timeouts specified for **Sendmail**:

```
O Timeout.datablock=XX
```

(the default value is 1 hour, XX=>1h).

After you make changes to `sendmail.cf` configuration file, recompile it.

`__ADDRESS__` string is a string that specifies the address of transport used to connect to **drweb-milter**. The string format and value are the same as those used in the **Address** parameter from the [Milter] [section](#) of the **Dr.Web MailD** configuration file.

For TCP-sockets address must be specified in the following format:

```
inet: __PORT__ @ __HOST__
```

where `__PORT__` and `__HOST__` must have definite values (e.g. `inet:3001@localhost`).

For UNIX sockets address must be specified in the following format:

```
local: __SOCKPATH__
```

where `__SOCKPATH__` string must define the path which is accessible with the privileges the filter is started (e.g. `local:/var/run/drweb-milter.sock`).

Additional information on filter configuration can be found in **Sendmail** system documentation. You must restart **Sendmail** after specifying all necessary parameter values.

To enable logging of **Sendmail** message identifiers by `drweb-maild` module (`sendmails` message ID) as well as sending to `drweb-maild` information on successful authorization, the following line must be included in `sendmail.cf`:

```
O Milter.macros.envfrom=i,{auth_type}, ...
```

(suspension points denote other parameters, which values are of no importance in this case).



To allow **Dr.Web MailD** to define IP address and host name of the sender as well as to transfer `drweb-maild` module the interface address which received the message, add the following line to `sendmail.cf` configuration file:

```
O Milter.macros.connect=_{if_addr}, ...
```

(suspension points denote other parameters, which values are of no importance in this case).

To disable output of the following messages to `syslog`:

```
X-Authentication-Warning: some.domain.com: drweb set sender to DrWeb-DAEMON@some.domain.com using -f
```

include the user with whose privileges `drweb-milter` is operating (`drweb` user by default) to the `trusted-users` list in `submit.cf` file. This can be done by adding the user to the list directly in `submit.cf` and `sendmail.cf` configuration files:

```
#####  
# Trusted users #  
#####  
Tdrweb
```

Or by adding the following line to the `submit.mc` file:

```
define(`confTRUSTED_USERS', `drweb')
```

Configuring Dr.Web MailD

All settings that manage operation of `drweb-milter` with **Sender** component are collected in the `[Sender]` and `[Milter]` sections of the **Dr.Web MailD** configuration file and are described in the following sections of the current Manual: [\[Sender\] section](#) and [\[Milter\] section](#).

Ensure that the `SecureHash` parameter value in the `[Sender]` [section](#) of the **Dr.Web MailD** configuration file is specified (arbitrary character string can be set as the parameter value, recommended length is more than or equal to 10 symbols). In addition, `Yes` value must be specified for the `UseSecureHash` parameter in the same section. These parameters configure addition of a special header `X-DrWeb-Hash` that prevents looping of a message when it is checked (as the message can be added to the incoming email queue again if it was modified during the check).

When **Dr.Web MailD** is working with **Sendmail**, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of the **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`
- `drweb-milter`

Known Issues

Description:

When UNIX socket is used for communication between the filter and **Sendmail**, `Milter` API library (distributed with **Sendmail**) could not remove (prior to version 8.12.2) the file used for the socket.

Solution:

For versions 8.12.x the following patch is available - `listener-8.12.0-1.patch`. For versions 8.11



and later, this file must be removed manually or via script which manages filter operation. The issue is resolved in **Sendmail** version 8.12.2

Description:

When demo key is used in the local scan mode, the `size` value of a message which is transmitted to the next server is doubled after passing through the filter (the message itself either remains unchanged or a small "banner" is added to it).

Solution:

This issue is resolved in **Sendmail** version 8.12.3 and later.

Description:

When the filter operates on computers with high load, the following entries can be found in the log:

```
... Milter (drweb-milter): select(read): interrupted system call
```

Solution:

This issue is resolved in **Sendmail** version 8.12.3 and later.

Description:

When the filter operates on computers with high load, the following entries can be found in the log:

```
... Milter (drweb-milter): select(read): timeout before data write
... Milter (drweb-milter): to error state
```

Solution:

The problem is that **Sendmail** cannot establish connection with the filter within the specified timeout. In versions 8.11 and later the timeout is set to 5 seconds and cannot be changed. In versions 8.12 and later this timeout can be changed in the description of the filter (the C value):

```
Xdrweb-milter, S=__ADDRESS__, F=T, T=C:1m;S:5m;R:5m;E:1h
```

Integration with Mail Postfix

Main Operation Principles

Dr.Web MailD can be connected to **Postfix** in one of the following ways:

- In the **after-queue** mode (http://www.postfix.org/FILTER_README.html#advanced_filter);
- In the **before-queue** mode (http://www.postfix.org/SMTDPD_PROXY_README.html);
- Using `Milter` protocol (http://www.postfix.org/MILTER_README.html).



Only **Postfix** version 2.3.3 and later can be used with `Milter` protocol.

Operation in After-Queue and Before-Queue Modes

In the **after-queue** mode **Dr.Web MailD** interacts with **Postfix** in the following way:

1. `drweb-receiver` [module](#) (**Receiver** component), acting as an SMTP/LMTP server, receives a



new message from **Postfix** SMTP module and redirects it to `drweb-maild` (**MailD core** component) for analysis.

2. According to the analysis results, the message is either sent to the mail system (maybe as a modified copy) or blocked (in this case, additional reports can be sent to the mail system).
3. Redirecting messages to **Postfix** mail system is performed via `drweb-sender` acting as an SMTP/LMTP client which dispatches messages to `smtpd` daemon.

For details on configuring filters for **Postfix**, refer to the **Postfix** documentation, which can be found, for example, at http://www.postfix.org/FILTER_README.html.

Dr.Web MailD can also interact with **Postfix** server in the **before-queue** mode as well (but it is not recommended to use this mode if system load is high). For more information on how to configure operation in **before-queue** mode, refer, for example, to the following web page: http://www.postfix.org/SMTPLD_PROXY_README.html.

Operation Using Milter Protocol

Interaction with **Postfix** system via `Milter` protocol is organized in the following way:

- Through the transport connection defined by the transport address of `drweb-milter` (which acts as a **Receiver component**) **Postfix** system receives internal commands from `Milter` API and a message itself. The message is transferred in segments depending on the stage of the mail session (`helo`, `mail from:`, `rcpt to:` etc.). These segments are saved by `drweb-milter` module to the temporary directory. Through `Milter` API, `drweb-milter` transmits instructions to the **Postfix** system about actions to be applied to the message.

`Milter` API is a multithreaded library, which allows to process several mail sessions simultaneously. In the interaction solution, described above, **Postfix** system is a client and `drweb-milter` is a server, therefore in the `mail.cf` configuration file of **Postfix** system it is required to specify address of `drweb-milter` module, and **Postfix** system selects the appropriate client address for this connection.

- Through another transport connection, `drweb-milter` module transfers `drweb-maild` module commands and waits for response.

In the solution given above, `drweb-milter` module works as a proxy between the **Postfix** system interface and `drweb-maild` module.

Note [features of operation](#) through `Milter` in the synchronous and asynchronous modes.



Postfix and `drweb-milter` can operate on different computers, but `drweb-milter` and `drweb-maild` must operate on the same computer.

Configuring Mail Postfix

Configuring Operation in After-Queue Mode

To configure interaction between **Dr.Web MailD** and **Postfix** in the **after-queue** mode, add the following lines to the `main.cf` configuration file of the **Postfix** system:

```
content_filter=scan:<_ADDR_REC_>
receive_override_options=no_address_mappings
```

where `_ADDR_REC_` is the address of the [listening module](#) `drweb-receiver` (the **Address**



parameter of the [Receiver] [section](#) of the **Dr.Web MailD** configuration file), for example, 127.0.0.1:8025.

To the `master.cf` configuration file of **Postfix** system, the following lines must be added:

```
scan unix - - n - <NN> smtp
  -o smtp_send_xforward_command=yes
<ADDR_SEN> inet n - n - <NN> smtpd
  -o content_filter=
  -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
  -o smtpd_helo_restrictions=
  -o smtpd_client_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

where `<ADDR_SEN>` is the address to which `drweb-sender` module is connected for sending messages (the **Address** parameter of the [Sender] [section](#) of the **Dr.Web MailD** configuration file), for example, 127.0.0.1:8026.

It is recommended that `<NN>` number (maximum number of processes executed by **Postfix** server) is the same as the number of threads in pools of `drweb-receiver` and `drweb-sender` modules (the **PoolOptions** parameter in the [Receiver] [section](#) and **OutPoolOptions** in the [Sender] [section](#) of the **Dr.Web MailD** configuration file). To remove this limitation, specify "-" (minus sign) instead of the `<NN>` number.



During installation of **Dr.Web for UNIX mail servers**, all of the described changes are made to **Postfix** configuration files automatically with the use of `configure_mta.sh` script. So, **Dr.Web for UNIX mail servers** and **Postfix** are set for operation in the **after-queue** mode by default.

After applying the changes to, restart **Postfix**.

Configuring Operation Using Milter Protocol



To operate in this mode, **Postfix** system version 2.3.3 or later is required.

By default, **Dr.Web for UNIX mail servers** and **Postfix** are configured to interact in the **after-queue** mode. So, new settings that configure operation via `Milter` protocol, must be specified in the **Postfix** configuration files instead of the existing ones: change the `content_filter` parameter to the `smtpd_milters` parameter and remove all of the changes made to `master.cf` file. Necessary restrictions can be specified directly in the **Postfix** configuration files.

Address of the transport connection through which **Postfix** interacts with `drweb-milter` module can be specified as a TCP socket or as a UNIX socket.

Address is specified in the `smtpd_milters` parameter of the **Postfix** configuration file `main.cf`. If the connection is established via a TCP socket, the parameter value is set in the following format: `inet:host@port` (for example, `smtpd_milters=inet:127.0.0.1:3001`). If the connection is established through the UNIX socket, the address is set in the following format: `unix:pathname`, where `pathname` is an absolute path to the UNIX socket.



If UNIX socket is used, **Postfix** must have privileges for writing to the socket file.

Address of the transport connection between **Postfix** system and `drweb-milter` module must be also specified in the **Address** parameter of the [Milter] [section](#) in **Dr.Web MailD** configuration file.



Format and value of this parameter must be identical to the format and value of `smtpd_milters` parameter in `main.cf` file.

Apart from transport address, the following parameters must be specified in `main.cf` configuration file:

- `milter_content_timeout = 300s` – this timeout of **Postfix** system is very important. It defines the maximum time period for **Dr.Web MailD** to check a message in the `BeforeQueueFilters` mode. It is recommended to set this parameter value greater than the value of the `ProcessingTimeout` parameter of the `[Milter]` [section](#) in the **Dr.Web MailD** configuration file;
- `milter_default_action = tempfail` – this parameter defines action of **Postfix** if any errors occur during interaction with `drweb-milter` module;
- `milter_protocol = 6` – the required version of `Milter` protocol;
- `milter_mail_macros = _` – this parameter allows **Dr.Web MailD** to retrieve the IP address and host name of the sender;
- `milter_end_of_data_macros = i {auth_type}` – this parameter allows to retrieve information on authorization and the message ID to add information on the message to `drweb-milter` log.

Note [features of operation](#) through `Milter` in the synchronous and asynchronous modes.

Configuring Dr.Web MailD

If the suite is started using **Dr. Web Monitor component**, `drweb-milter` module must be started as a **Receiver** component. For that, in the `%etc_dir/monitor/maild_postfix.mmc` file uncomment the string [which is responsible for startup](#) of `drweb-milter` module. It is also recommended to comment the string which is responsible for startup of `drweb-receiver` module. As a result, `drweb_postfix.mmc` contains strings similar to the following ones:

```
# drweb-receiver local:%var_dir/ipc/.agent 15 30 MAIL drweb:drweb
drweb-milter local:%var_dir/ipc/.agent 15 30 MAIL drweb:drweb
```

It is also required to configure operation of `drweb-sender` module. Specify the following parameters in the `[Sender]` [section](#) of the **Dr.Web MailD** configuration file:

```
Address = /usr/local/sbin/sendmail
Method = pipe
MailerName = postfix
```

In the `Address` parameter, you can set the path to the `sendmail` program from **Postfix** package.

Once all the required parameters are specified, you must start/restart **Dr.Web MailD** at first, and then start/restart **Postfix**.

All settings that configure operation of `drweb-milter` with **Sender** and **Receiver** components are collected in the `[Receiver]`, `[Sender]` and `[Milter]` sections of the **Dr.Web MailD** configuration file and are described in the following sections of the current Manual: [\[Receiver\] section](#), [\[Sender\] section](#) and [\[Milter\] section](#).

When **Dr.Web MailD** is working with **Postfix** mail system, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`



- `drweb-receiver`

Integration with Exim



Integration instructions in this documentation are given for **Exim** version 4.xx. If you want to use earlier versions of **Exim** (3.xx), please refer to the corresponding documentation (for example, <http://www.exim.org/index.html>).

When **Dr.Web MailD** interacts with **Exim** mail system, `drweb-receiver` module acts as a **Receiver** component, and `drweb-sender` acts as a **Sender** component. There are two possible methods to connect **Exim** mail system to **Dr.Web MailD**:

- Connection by the means of special transport.
Advantages: recompilation of **Exim** is not required, and system can operate with relatively old versions of **Exim**.
Disadvantages: system performance is reduced.
- Connection by the means of **Exim** `local_scan` function. In this case, **Receiver**, unlike other components, receives its configuration data from the configuration file of **Exim** mail system (unlike other components, which receive configuration settings from **Dr.Web Agent component**).
Advantages: system performance is increased.
Disadvantages: recompilation of **Exim** is required, and **Exim** version must be 4.50 or later.



Note that the initialization script included in **Dr.Web MailD** (and located in the `%bin_dir/maild/scripts/` directory) allows to preconfigure **Dr.Web MailD** and **Exim** to interact [using special transport](#).

Configuring Exim

Initial configuration is identical for both connection methods.

First, it is necessary to add `drweb` user to the list of trusted users in the `MAIN CONFIGURATION SETTINGS` section of the **Exim** configuration file:

```
#####  
#           MAIN CONFIGURATION SETTINGS           #  
#####  
trusted_users = drweb
```

Please note that if **Exim** performs mail delivery immediately after receipt of messages from `drweb-sender`, and serious delays occur in this delivery (e.g., when SMTP protocol is used), then timeout specified as the `PipeTimeout` parameter value in the `[Sender]` [section](#) of the **Dr.Web MailD** configuration file can be applied, because **Exim** does not return the code of successful receipt to `drweb-sender` until the delivery is finished. To avoid this problem, you may configure **Exim** to send all messages to the queue first, and only after that - to perform delivery.



Add a new `acl` to the **Exim** configuration file:

```
acl_check_drweb_scanned:
warn
condition = ${if and {{def:received_protocol}{eq ${received_protocol}}\
{drweb-scanned}}} {yes}{no}}
control = queue_only
accept
```

and then enable it:

```
acl_not_smtp = acl_check_drweb_scanned
```

Connecting to Exim Using Special Transport



The description below is valid only for **Exim** 4.xx. For information on adjusting the settings for earlier versions of **Exim** (3.xx), refer to the corresponding documentation (for example, at <http://www.exim.org/index.html>).

In **Exim** settings you must add a special transport and a router. Find the `Routers Configuration` section in the configuration file of the mail system. It starts with the following header:

```
#####
#          ROUTERS CONFIGURATION          #
# Specifies how remote addresses are handled #
#####
#          ORDER DOES MATTER              #
# A remote address is passed to each in    #
#      turn until it is accepted.          #
#####
```

And right after the following line

```
begin routers
```

add the following description:

```
drweb_router:
  driver = accept
  condition = "${if eq ${received_protocol}{drweb-scanned}{0}{1}}"
# check_local_user
  retry_use_local_part
  transport = drweb_transport
```

If check of the recipients is necessary, uncomment the `check_local_user` parameter.



In the **Exim** configuration file, find the section where transport is described. It starts with the following header:

```
#####  
#      TRANSPORTS CONFIGURATION      #  
#####  
#      ORDER DOES NOT MATTER      #  
# Only one appropriate transport is called #  
#      for each delivery.      #  
#####
```

You must add a description of the required transport to this section:

```
drweb_transport:  
  driver = lmtp  
  socket = __ADDRESS__  
  batch_max = 100  
  timeout = 5m  
  user = drweb  
# headers_add = "X-Maild-Checked: DrWEB for Exim"
```

Where `__ADDRESS__` is the address of **drweb-receiver** [listening module](#) (the **Address** parameter in the `[Receiver]` [section](#) of the **Dr.Web MailD** configuration file) – for example, a UNIX socket `%var_dir/ipc/.drweb_maild`.

Then you must specify the path to **Exim** mail system in the **Address** parameter in the `[Sender]` [section](#) of the **Dr.Web MailD** configuration file (for example, `/usr/exim/bin/exim/`), and specify **Exim** as a value of the **MailerName** parameter from the same `[Sender]` [section](#).

Connecting to Exim Using Local_Scan Function



Working with **Dr.Web MailD** in this mode requires **Exim** mail system version 4.50 or later.

Note that the steps describe how to configure connection using the `local_scan` function assuming that the **Exim** configuration file was not changed; that is, the file does not contain adjustments for using special transport, described [in the previous section](#). Thus, if the file was previously preconfigured for that purpose, it is required to revert the **Exim** configuration file to its initial state by removing settings for the router and transport.

Preparation of the system has several stages. First, you must recompile **Exim** with support of the `local_scan` function:

- Copy `%bin_dir/doc/maild/local_scan/local_scan.c` file to `exim*/Local/` directory.
- To Makefile of **Exim** system, which is located in `exim*/Local/` directory, add parameters specified in `%bin_dir/doc/maild/local_scan/Makefile.sample`. If the corresponding parameters are already specified in `Makefile`, you can uncomment and edit them.
- In `Makefile` of **Exim** system you must also specify the name of a user whose privileges are used to start **Exim** (the name must be the same as specified for **Dr.Web MailD**). User name is defined by the `EXIM_USER` parameter. By default, `EXIM_USER = drweb`.



- Compile and install **Exim** system. If the execution of `make` or `make install` commands is interrupted with error messages like:

```
/libexec/ld-elf.so.1: Shared object "libgcc_s.so.1" not found, required by "libboost_thread.so"
```

then there are two possible ways to fix it:

- copy `libstdc++.so.6` and `libgcc_s.so.1` libraries (or make links with the same name to them) from `%bin_dir/lib/` to the system library directory.
- execute the following command from the console:

```
$ export LD_LIBRARY_PATH=%bin_dir/lib/:$LD_LIBRARY_PATH
```

and then compile and install **Exim** once again from the console.

After that, configure **Exim** system. For quick configuration you may use values of parameters from `%bin_dir/doc/maild/local_scan/configure.sample` file. Just copy the necessary lines to the `local_scan` section of the **Exim** configuration file.

To retrieve information on the **Receiver** settings, execute the following command from the console:

```
$ PATH_TO_BIN_DIR/exim -bP local_scan
```

where `PATH_TO_BIN_DIR` is the path to the **Exim** binaries.

In the **Exim** configure file, the following additional parameters can be set:

<code>DrwebTimeout = {time}</code>	<p>Period during which Exim is waiting for <code>drweb-maild</code> to scan a message.</p> <p>It is recommended to set a greater value than the one of the SendTimeout parameter in the [MailBase] section.</p> <p><u>Default value:</u> DrwebTimeout = 60s</p>
<code>DrwebBaseDir = {logical}</code>	<p>Dr.Web MailD base directory where sockets, database etc. are stored</p> <p><u>Default value:</u> DrwebBaseDir = %var_dir/</p>
<code>DrwebProcessingError = {action}</code>	<p>This parameter defines what action is to be applied to messages which caused scanning errors (for example, if the anti-virus plug-in runs short of memory or cannot connect to <code>drweb-maild</code>).</p> <p>Allowed actions: pass, discard, reject, tempfail</p> <p>If the parameter value is not set or several values are mistakenly specified (for example, discard and pass) – tempfail value is used by default.</p> <p><u>Default value:</u> DrwebProcessingError = tempfail</p>
<code>DrwebLogLevel = {log level}</code>	<p>Log verbosity level.</p> <p>The following levels are allowed:</p> <ul style="list-style-type: none">• Quiet• Error• Alert• Info• Debug



	<p>Default value: DrwebLogLevel = Debug</p>
<p>DrwebIpcLevel = {log level}</p>	<p>IPC library log verbosity level. The following levels are allowed:</p> <ul style="list-style-type: none"> • Quiet • Error • Alert • Info • Debug <p>Default value: DrwebLogLevel = Debug</p>
<p>DrwebSyslogFacility = {syslog label}</p>	<p>Type of facility which generates a notification message on event when using syslog system service.</p> <p>Default value: DrwebSyslogFacility = Daemon</p>
<p>DrwebMaxSize = {size}</p>	<p>Maximum size of a checked message. When 0 is specified, size is not limited.</p> <p>Default value: DrwebMaxSize = 200 k</p>

Configuring Dr.Web MailD

To configure **Dr.Web MailD** for interaction with **Exim**, specify the path to the **Exim** mail system in the **Address** parameter in the [Sender] section of **Dr.Web MailD** configuration file (for example, /usr/exim/bin/exim/) and set **Exim** as a value of the **MailerName** parameter from the same section.

There is no need to start **drweb-receiver** [module](#) separately, because when operation through **local_scan** function is performed, **Receiver** component is embedded into **Exim**. If **Dr.Web MailD** is started using **Dr.Web Monitor**, comment the line in `%etc_dir/monitor/maild_exim.mmc` [file](#), which is responsible for **drweb-receiver** startup:

```
#drweb-receiver local:%var_dir/ipc/.agent 15 30 MAIL drweb:drweb
```

After all changes are made, restart **Dr.Web MailD** and **Exim** mail system.

All settings that configure operation of **Dr.Web MailD** with **Exim** are collected in the [Receiver] and [Sender] sections of the **Dr.Web MailD** configuration file and are described in the following sections of the current Manual: [\[Receiver\] section](#) and [\[Sender\] section](#).

When **Dr.Web MailD** is interacting with the **Exim** mail system, the following processes must be running:

- **drweb-notifier**
- **drweb-sender**
- **drweb-maild**
- **drweb-receiver**

Also it is additionally necessary to make, that the **Dr.Web Monitor** [component](#) has been launched with the `root` privileges (to provide this, specify `root` value for **User** and **Group** parameters in the



[Monitor] [section](#) of `monitor.conf` configuration file).

Known Issues

If on **Exim** startup, the following error occurs:

```
transport drweb_transport: cannot find transport driver "lmtp"
```

it means that **Exim** is built without LMTP transport support. You can either switch to SMTP transport (for details, refer to documentation on **Exim** MTA, for example, at <http://www.exim.org/>), or recompile **Exim** with LMTP transport support.

If the latter variant is used, you must add or uncomment the following line in `/Local/Makefile` file of the **Exim** system: `TRANSPORT_LMTP = yes`.

Integration with Qmail

Operation principle of **Qmail** is based on replacement (proxying) of the mail system. Via the interface set for `qmail-queue` module (the main executable file of **Qmail** system), the filter receives a message, checks it and if the message is not infected, the filter moves it to `qmail-queue`.

Operation in this mode has the following limitation: UNIX sockets, which `drweb-qmail` listens for scan requests (are set in the `ListenUNIXSockets` parameter in the `[Qmail]` [section](#) of the **Dr.Web MailD** configuration file) must be located in certain directories. To display a list of paths, run the `qmail-queue` with `--help` command line parameter).



For integration with **Dr.Web MailD Qmail** version not earlier than 1.03 is required. To avoid possible loss of incoming mail, filter must be installed only when **Qmail** is stopped.

Integration of **Dr.Web MailD** with **Qmail** can be performed both manually (refer to the [instructions](#) below) and with the use of the `configure_mta.sh` configuration script (that resides in the directory `%bin_dir/maild/scripts/`). It can configure interaction, as mentioned in the instructions below, but cannot add users to the corresponding groups. Thus, if you used the script for integration configuration, do not forget to add users to the groups (see below).

Configuring Qmail

To connect **Dr.Web MailD** to **Qmail** mail system, do the following:

1. Open the directory with **Qmail** executable files `<qmail_dir>` (usually they reside in the `/var/qmail/bin` directory) and rename the `qmail-queue` file to `qmail-queue.original`;



Note that if you moved the `qmail-queue` original file to another location or defined another name (different from `qmail-queue.original`), it is required to specify a new path to this file in the [main configuration file](#) of **Dr.Web MailD**. To do this, set the new file path as a value of the `QmailQueue` parameter in the `[Qmail]` [section](#).



2. Instead of the renamed file create the following symbolic link in the same directory: `qmail-queue`
`-> %bin_dir/qmail-queue;`
3. Set rights for users whose privileges are used to run the files.

The most convenient configuration is the one when **Dr.Web MailD** and `qmail-queue` operate under the `drweb` account. To enable proper functioning of the configuration, assign the following rights for `%bin_dir/qmail-queue` and `%qmail_dir/qmail-queue`:

```
-rws--x--x X drweb qmail SIZE DATE %bin_dir/qmail-queue
-rws--x--x X qmailq qmail SIZE DATE %qmail_dir/qmail-queue.original
```

You can also do that using the following commands:

```
# chown drweb:qmail %bin_dir/qmail-queue
# chmod 4711 %bin_dir/qmail-queue
# chown qmailq:qmail %qmail_dir/qmail-queue.original
# chmod 4711 %qmail_dir/qmail-queue.original
```

4. Also add `qmailq` and `qmaild` users to the `drweb` group and `drweb` user to the `qmail` group.

Configuring Dr.Web MailD

All settings providing proper operation of **Dr.Web MailD** with **Qmail** are collected in the `[Sender]` and `[Qmail]` sections of **Dr.Web MailD** configuration file and are described in the following sections of the current Manual: [\[Sender\]](#) and [\[Qmail\]](#).

It is recommended to set the `SecureHash` parameter value in the `[Sender]` [section](#) of the **Dr.Web MailD** configuration file (you can specify a free-form string, advisory length – not less than 10 characters) and set the `UseSecureHash` parameter value in the same section to `Yes`. These parameters configure addition of a special header `X-DrWeb-Hash` that prevents looping of a message when it is checked (as the message can be added to the incoming email queue again if it was modified during the check).

When **Dr.Web MailD** work with **Qmail** system, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`
- `drweb-qmail`



Known Issues

Description:

On startup, **Qmail** returns one of the following errors:

```
terminate called after throwing an instance of 'St9bad_alloc'  
what(): St9bad_alloc
```

```
bash: xmalloc: cannot allocate 2 bytes (0 bytes allocated)
```

```
qmail-queue.real: error while loading shared libraries: libc.so.6: failed to  
map segment from shared object:  
Cannot allocate memory
```

```
/var/qmail/bin/qmail-smtpd:  
error while loading shared libraries:  
libc.so.6: failed to map segment from shared object:  
Cannot allocate memory
```

Solution:

The problem occurs because of memory usage limit in the initialization script is too high. For example, if Dave Sill scripts are used, the value indicated in the instruction `softlimit -m 2000000` must be increased by adding, for instance, to `200000000`.

Description:

In reply to all messages received via SMTP-protocol **Qmail** returns the following string:

```
451 qq trouble making network connection (#4.3.0)
```

Solution

`qmail-queue` can have not enough privileges to connect to the UNIX socket, created by `drweb-qmail` (which [operates](#) as **Receiver** component of **Dr.Web MailD**), or paths specified by default for `qmail-queue` do not lead to this socket. Check the privileges and ensure that the value of the `ListenUNIXSocket` parameter in the [Qmail] [section](#) of the **Dr.Web MailD** configuration file matches the default paths (a list of these paths can be obtained by running `qmail-queue` with the `--help` command line parameter).

Description:

For each message received via SMTP-protocol **Qmail** outputs the following string to the console upon receipt of a message body:

```
qmail-inject: fatal: qq temporary problem (#4.3.0)  
/usr/libexec/ld-elf.so.1: Shared object "libstdc++.so.6" not found,  
required by "libboost_program_options.so"
```

Solution:

The system cannot find necessary libraries which are located in `%bin_dir/lib/` directory. It is necessary to copy (or create a symbolic link) `libstdc++.so.6` and `libgcc_s.so.1` from `%bin_dir/lib/` to the system library directory.



Integration with ZMailer



`drweb-zmailer` module is compatible only with **ZMailer** v. 2.99.55 or later.

Dr.Web MailD can interact with **ZMailer** in two modes:

- As a content filter at the stage of SMTP-connection.

Advantages: it is possible to block the message at the stage of SMTP-connection.

Disadvantages: decreased performance when system load is high, only SMTP traffic is checked.

- As a content filter at the routing stage.

Advantages: stable performance when system load is high; all mail coming through **ZMailer** is checked (including local mail and mail transferred via UUCP protocol).

Disadvantages: message cannot be blocked when it is received (i.e., `reject` and `tempfail` actions are similar to `discard`); usage of `SecureHash` is necessary to increase performance and avoid cycling of messages.

`drweb-zmailer` [module](#) is used as **Receiver** component of **Dr.Web MailD** when interacting with **ZMailer**.

To assure proper operation of `drweb-zmailer` and filters it is recommended to install patches (if possible).

To install patches, do the following:

- Open the directory,

```
$ (ZMAILER_SRCHOME) /smtpserver
```

where `ZMAILER_SRCHOME` is the path to the directory with **ZMailer** binaries

- Run the following command:

```
$ patch < smtpdata.c.XXX.patch
```

where `XXX` stands for the version of **Zmailer** to be patched.

When **Dr.Web MailD** is interacting with **Zmailer** system, the following [modules](#) must be running in the system (this is specified in [mmc file](#) of the **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`

Content Filter at SMTP Session Stage

To enable **Dr.Web MailD** support in **ZMailer**, do the following:

- copy (or make a symbolic link) `drweb-zmailer.sh` to `$MAILBIN` directory (path to the file is specified in `zmailer.conf`);
- edit `smtpserver.conf` file by adding the following string (or modifying the existing one):

```
PARAM contentfilter $MAILBIN/drweb-zmailer.sh.
```

As command line parameters cannot be specified in `contentfilter`, you must define them in



`drweb-zmailer.sh` script.

Content Filter at Routing Stage

All messages processed by the mail server pass the routing stage. Therefore, the end of the routing stage is the most suitable time for the filter to connect. To make such connection possible, edit `MAILBIN/cf/process.cf` file as described below:

Find the following lines:

```
LOGMSG=() # This is a LIST of files where to log..
#| The LOGMSG variable is used by the intercept facility (in crossbar.cf)
#| to make sure only a single copy of a message is saved when required.
#| Each sender - recipient address pair can cause an intercept which can
#| specify a file to save the message to. This variable is appended to
#| elsewhere, and processed at the end of this function.
```

Add the following:

```
###-> Dr.Web MailD support
ch=""DEFAULT_BIN_PATH/drweb-zmailer.sh" --hash __EDIT_THIS__ --file
$POSTOFFICE/router/$file'
  case "$ch" in
    -1*) #reject or disacrd
        /bin/rm -f "$file"
        return
        ;;
    1*) #tempfail
        /bin/rm -f "$file"
        return
        ;;
    *) ;;
  esac
###-> end of Dr.Web MailD support
```

Replace `__EDIT_THIS__` (value of the `--hash` [parameter](#)) with the value equal to the one of the `SecureHash` parameter in the `[Sender]` [section](#) of **Dr.Web MailD** configuration file and specify `Yes` for the `UseSecureHash` parameter in the same section.

Additional Settings

A simple way to disable receipt of messages with an empty SMTP sender envelope (usually error messages or messages saying that delivery failed are sent with an empty SMTP envelope; also such messages are often sent by spammers) is to install `policytest.c.XXX.patch`. Installation procedure is similar to installing `smtpdata.c.XXX.patch` file.

As **ZMailer** starts `drweb-zmailer` module each time a new message is processed, for optimized system performance all `drweb-zmailer` settings must be specified in the command line (they can be defined, for example, in `drweb-zmailer.sh` script).

Parameters that can be specified in the command line of `drweb-zmailer` module are described in [Command line parameters](#) section.

Integration with Courier



Configuring Courier

To connect **Dr.Web MailD** to **Courier** mail system, do the following:

1. Set privileges for **drweb-courier** [module](#) with the following commands:

```
$ chown COURIER_USER:drweb "%bin_dir/drweb-courier"  
$ chmod 6771 "%bin_dir/drweb-courier"
```

where `COURIER_USER` is a user with whose privileges **Courier** is started. Also ensure that read, write and execution permissions are set for all directories and subdirectories in `%var_dir` directory for the `drweb` group.

2. Copy **drweb-courier** module (or create a symbolic link) to **Courier** filters directory (by default, it is `/usr/local/libexec/filters/`).
3. Register **drweb-courier** module in the **Courier** mail system as global:

```
$ /usr/local/sbin/filterctl start drweb-courier
```

Later, to disable the filter, use the following command:

```
$ /usr/local/sbin/filterctl stop drweb-courier
```

4. Create (or edit) `enablefiltering` file to set services to perform check (`esmtplib` or `uucplib` - if more than one is specified, they are separated with white spaces).
5. Ensure that the `BaseDir` and `SocketDirs` parameters in the `[Courier]` [section](#) of the **Dr.Web MailD** configuration file correspond to the configuration of the used **Courier** mail system. For more information, use the following command: `man courierfilter`.
6. Enable filtering in **Courier** system:

```
$ /usr/lib/courier/sbin/courierfilter start
```

`drweb` user with whose privileges **Dr.Web Daemon** [component](#) operates, must be included in `courier` group to gain read access to files created in the spool by **Courier** mail system.

Transmission of processed messages to Courier MTA

Settings that manage transmission of processed messages to MTA are specified in the `[Sender]` [section](#) of the configuration file. For that purpose, the following parameters must be specified:

```
MailerName = Courier  
Method = pipe  
Address = <path to the post system for message sending>  
#(by default, /usr/lib/courier/bin/sendmail)
```

Configuring Dr.Web MailD

All settings providing proper operation of **Dr.Web MailD** with **Courier** are collected in the `[Sender]` and `[Courier]` sections of the **Dr.Web MailD** configuration file and described in the following sections of the current Manual: [\[Sender\] section](#) and [\[Courier\] section](#).

When **Dr.Web MailD** is working with mail system **Courier** the following [modules](#) must be running in the system (this is specified in [mmc file](#) of **Dr.Web Monitor**):

- `drweb-notifier`
- `drweb-sender`
- `drweb-maild`



- `drweb-courier`

Known Issues

Description:

During **Dr.Web MailD** restart due to receipt of `SIGHUP` [signal](#), the following problems can occur on high load, or when one of **Dr.Web Daemon** [scanning daemons](#) is not available (addresses of which are listed in **Drweb** plug-in settings), or if the scanning timeout, specified in the plug-in settings (the `Timeout` parameter), is too large: **MailD core** and **Notifier** components return an error and **Dr.Web Monitor** [component](#) restarts these components and send a notification email message to the administrator. The record in **Dr.Web Monitor** log file is as follows (for example):

```
monitor ERROR component "drweb-maild" terminated by signal 6 (Aborted)
monitor DEBUG component "drweb-maild" cannot stop
monitor DEBUG send notification From:<email@address>#012To:<email@address>.
Command: /usr/sbin/sendmail -t
```

Solution:

The error occurs because **MailD core** and **Notifier** components create a large number of active threads for traffic processing and upon receipt of `SIGHUP` signal the threads are not able to terminate correctly during a time period specified in the `MaxTimeoutForThreadActivity` [parameter](#) value. That is a sign that the hardware-and-software complex, which is processing mail traffic, is overloaded (at least - at traffic peaks). To solve the problem

- Increase timeout value in the `MaxTimeoutForThreadActivity` parameter or limit the number of active threads in pools of these components.

It is strongly recommended to take the following actions for stable operation of the software suite:

- Estimate the volume of processed traffic at its peaks (for that purpose, use log files and statistics)
- [Optimize](#) usage of **Dr.Web MailD** system resources;
- If optimization is not possible or it does not bring the desired result, update the hardware (increase RAM memory and number of available CPU cores).

As a temporary solution, you can define the `DW_FORCE_EXIT` environmental variable (set any value). In this case, **Dr.Web Monitor** does not send notifications to the administrator if the components stop responding upon their shutdown and terminates operation of the whole complex and after that shuts down (used for compatibility with earlier **Dr.Web MailD** versions).

Description:

On **Dr.Web MailD** startup, **Dr.Web Monitor** abnormally terminates **MailD core** operation and send an email notification to the administrator. The record in the **Dr.Web Monitor** log is as follows (for example):

```
monitor DEBUG DEBUG component "drweb-maild" not answer
monitor DEBUG Component::stopPid() # drweb-maild <comp name="drweb-maild"
argv="local:/var/drweb/ipc/.agent" start="120" stop="30" log="2"
user="drweb:drweb" fd="-1" pid="7194"/>
monitor DEBUG kill -TERM pid=7194 name="drweb-maild"
```

Solution:

The error occurs because **MailD core** creates too many threads and the component requires more time to start than it is specified in **Dr.Web Monitor** [startup settings](#). To solve the problem

- Restrict the minimum number of active threads in **MailD core** pools.

It is strongly recommended to do the following in order to provide stable operation of the software suite:



- Estimate the amount of mail traffic at peak usage time (use logs and statistics).
- [Optimize](#)[****]**Dr.Web MailD** operation and system resources usage.
- If it cannot be optimized or actions performed on the previous step did not have the intended effect, modernize the hardware (increase RAM and number of available processor cores).

Description:

On **Solaris** OS, **%MAILD%>** registers WARN messages in the log when generating notifications on a processed message or during message processing. The messages are of the following type:

```
notifier WARN Decoding string ' \362\345\361\362
notifier WARN because of iconv error: Invalid argument
```

Solution:

To solve the problem, change the version of the `iconv` system envelope and use `iconv` from the `libiconv` package, which is distributed under GNU license (the package can be downloaded at <http://www.gnu.org/software/libiconv/#downloading>).

Description:

When using 32-bit **Solaris** OS version 10, the following problem can occur if load on **Dr.Web MailD** is high:

Incoming messages cannot be accepted for processing and the following message is fixed in the log file: "Too many open files".

Solution:

The error occurs because the number of open file (and socket) descriptors is exhausted. To solve the problem, [optimize](#) operation and system resources usage.

Description:

Cannot connect to **MySQL** [data sources](#), **Dr.Web MailD** writes the following message to the log:

```
Cannot load library: Cannot load shared library libmysqlclient_r.so.18
because libmysqlclient_r.so.18: Undefined symbol "strlen"
```

Solution:

This error is detected only on **FreeBSD** OS, and if the solution includes `libc.so.7` library in the `/usr/local/drweb/lib64` directory.

To solve the problem, replace the `libc.so.7` library in the `/usr/local/drweb/lib64` directory to a similarly-named symbolic link to the system library `/lib/libc.so.7`.

Description:

In the **Dr.Web MailD** log, the following messages are registered:

```
Can not send msg from temp dir ('<dir_path>') -> remove dir and forget
about it
```

where `<dir_path>` - path to a directory in the message storage (for example, `/var/drweb/msgs/db/A/00000B9A`).

Solution:

The situation described above does not indicate an error in message processing. Such messages can be logged due to a repeated attempt to send messages when **Dr.Web MailD** is restarted after an emergency shutdown. These lines in the log are indicators that messages were successfully sent to the target MTA, but **MailD core**, the central component, did not finish clearing directories in the message



storage.

Description:

Users of **MS Exchange** mail server receive unreadable DSN if messages sent by them were not delivered.

Solution:

The situation described above is related to features of **MS Exchange** that is not fully compliant with RFC 3464 requirements. To solve the problem, replace the standard `dsn.msg` - DSN notification template to the special version designed specially for **MS Exchange**. The template is located in the `dsn_for_exchange.msg` file.

For details on how to replace the template, see [Notification processing](#).



Dr.Web Console for UNIX mail servers

Setup and configuration of **Dr.Web for UNIX mail servers** can be performed via a specially designed web interface **Dr.Web Console for UNIX mail servers**. It is implemented as a plug-in to **Webmin** (for details on the **Webmin** interface, visit its official website at <http://www.webmin.com/>).

To achieve optimal performance of the **Dr.Web Console for UNIX mail servers** web interface, ensure that the following **Perl** modules are installed in your system:

- **XML::Parser** – Perl module for parsing XML documents;
- **XML::XPath** – set of modules for parsing XPath statements;
- **Encode** – module for encoding conversion;
- **Date::Parse** – module for conversion date to UNIX format;
- **CGI** – module that enables operation with Common Gateway Interface;
- **CGI::Carp** – module for creation of HTTPD report on errors;
- **JSON** – module for parsing and converting to JSON (JavaScript Object Notation)
- **Digest::MD5** – module for using MD5 encryption algorithm;
- **MIME::Words** – module for using RFC 2047 encoding;
- **MIME::Entity** – module for decoding and parsing MIME messages;
- **MIME::Parser** – Perl module for parsing MIME threads;
- **MIME::Head** – Perl module for parsing headers of MIME messages;
- **File::Stat** – module with interface to embedded stat() functions;
- **File::Find** – module with interface to perform search through directory tree;
- **Encode::CN** – module used for Chinese character encoding;
- **Encode::HanExtra** – extra sets of Chinese encodings;
- **Switch** – module for `switch-case` statement usage.

If some modules are missing, it is recommended to install them from the command line. For that, `root` privileges are required. Names of the modules can differ, but they are usually included into the following packages: `perl-Convert-BinHex`, `perl-IO-stringy`, `perl-MIME-tools`, `perl-XML-Parser`, `perl-XML-XPath`. To install the modules in `rpm` systems, it is recommended to choose `noarch.rpm` packages.

Web interface layout and appearance can differ depending on the **Webmin** version and the used browser.



Due to peculiarities of **Webmin** implementation, **Dr.Web Console for UNIX mail servers** web interface cannot be correctly displayed in **Internet Explorer 7**. If problems with display of webpages occur, try to use **Internet Explorer 8** or **9** (or later) or use another browser.

Installation

To start working with **Dr.Web Console for UNIX mail servers**, do the following:

- install **Webmin**;
- install **Dr.Web Console for UNIX mail servers** plug-in located in `%bin_dir/web/`.

Webmin configuration and installation of modules is performed with the use of **Webmin** web



interface.

The screenshot shows the Webmin main page. On the left is a navigation menu with 'Webmin', 'Webmin Configuration', 'Servers', 'System Information', 'Refresh Modules', and 'Logout'. The main area features the Webmin logo and a system status table:

System hostname	
Operating system	Ubuntu Linux 9.04
Webmin version	1.480
Time on system	Wed Aug 26 16:53:16 2009
Kernel and CPU	Linux 2.6.28-15-generic on i686
System uptime	2 days, 4 hours, 21 minutes
CPU load averages	0.13 (1 min) 0.18 (5 mins) 0.28 (15 mins)
Real memory	1002.62 MB total, 654.62 MB used
Virtual memory	2.86 GB total, 478.04 MB used
Local disk space	180.56 GB total, 15.51 GB used

Figure 23. Webmin main page

To install additional modules, click **Webmin Configuration** on the main menu and then click **Webmin Modules** on the open page.

The screenshot shows the 'Webmin Configuration' page. The 'Webmin Modules' icon is highlighted with a red box. Below the grid of icons are several control buttons and their descriptions:

- Start at boot time**: Yes No. Change this option to control whether Webmin is started at boot time or not. If it is not currently started at boot and Yes is chosen, a new init script will be created.
- Restart Webmin**: Click this button to re-start the Webmin server process. This may be necessary if you have recently upgraded Perl.
- Submit OS Information**: Clicking this button will send information about your operating system and Perl version to the Webmin developers. This data will be strictly anonymous, and will provide information about which operating systems to best focus the development of Webmin on.
- Refresh Modules**: Re-check all Webmin modules for installed servers, and update those that appear in the 'Un-used modules' category.

Figure 24. Webmin configuration

To install required modules

1. Click the **Browse** button near the **From local file** text field on the **Webmin Modules** page. A new browser window opens to provide navigation through folders and files.
2. Choose the corresponding installation package from the list (%bin_dir/web/drweb-maild-web.wbm.gz by default).

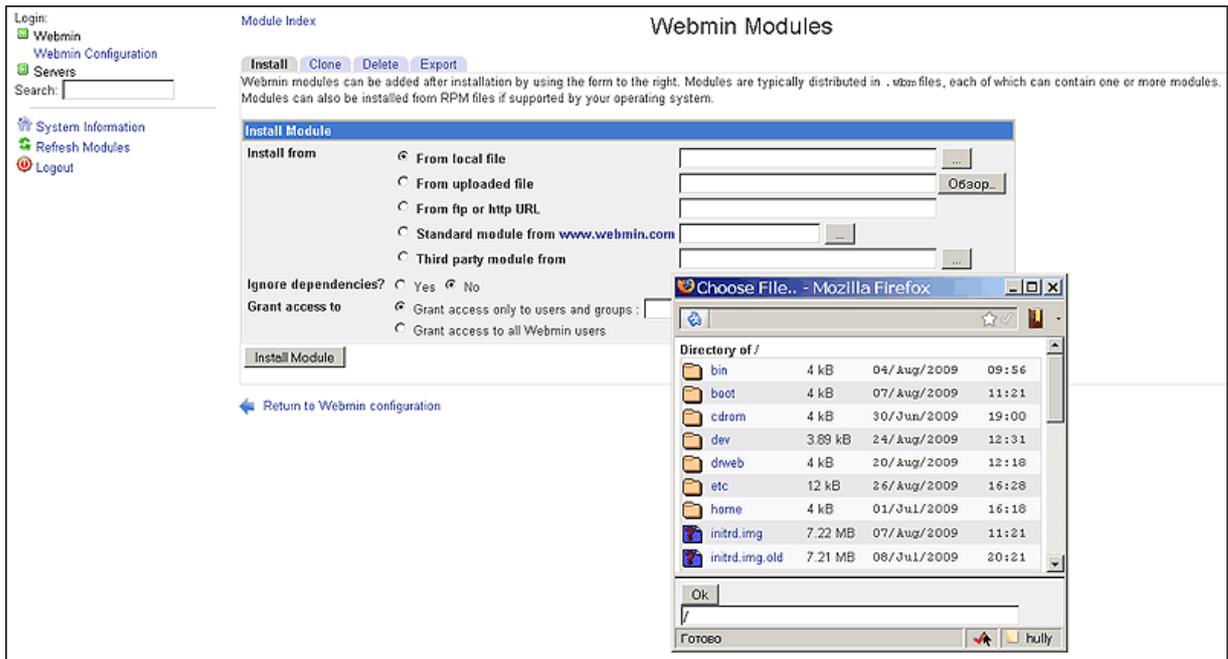


Figure 25. Webmin modules

3. After you click an item from the list, path to this item is added to the field below. If you click the item twice, the folder opens. With the second click on the previously selected file, navigation window closes, and the full path to the selected file appears in **From local file text** field. You may also click **OK** after you select a required file.
4. After you select an installation package file, click **Install Module**.
5. When the installation completes, a link to the new **Dr.Web Console for UNIX mail servers** module appears in the **Servers** section of the main menu.

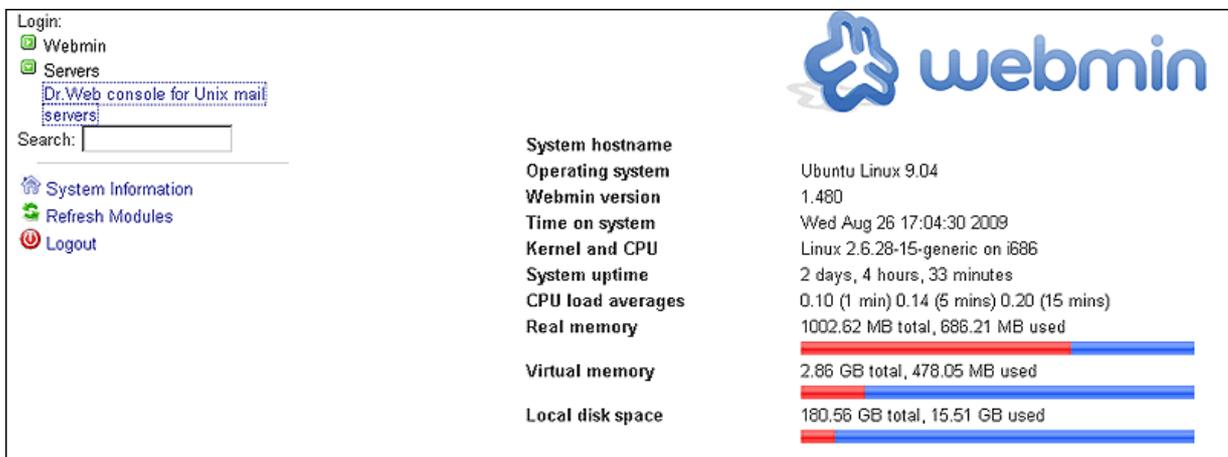


Figure 26. Dr.Web console for UNIX Mail Servers module



If you use **Webmin** 1.680 or later, you should also add the following line to its configuration file (usually, this is the file `/etc/webmin/config`):

```
no_content_security_policy=1
```

Basic Configuration

To open general settings of **Dr.Web Console for UNIX mail servers**, click **Module configuration** link on the top pane of the corresponding tab. On the open page, you can specify the used mail system,



path to the configuration file, path to the init script and mail sending script, default email address inserted to the **From** field of notifications on [Dr.Web for UNIX mail servers](#) and [operation mode](#).

Configuration

For module Dr.Web console for Unix mail servers

Configurable options for Dr.Web console for Unix mail servers

Dr.Web console for Unix mail server settings

MailD MTA	<input type="text" value="postfix"/>
MailD platform	<input type="text" value="linux"/>
Path to directory containing XML configuration files	<input type="text" value="/usr/share/webmin/drweb-m"/>
Maild config full path	<input type="text" value="/etc/drweb/maild_postfix.cc"/>
Path and arguments to script for sending emails	<input type="text" value="/opt/drweb/drweb-inject -f <"/>
Default section in Configuration	<input type="text" value="Basic"/>

Dr.Web Mail Daemon settings

Path to Maild installation	<input type="text"/>
Full path to MailD binaries	<input type="text" value="/opt/drweb"/>
Full path to MailD control (start/stop) script	<input type="text" value="/etc/init.d/drweb-monitor"/>

Interface settings

send emails from	<input type="text" value="maild"/>
Central protection mode	<input type="text" value="no"/>

[Return to index](#)

Figure 27. Module configuration



Do not forget to change the default value of the send emails from field. Otherwise, messages sent from **Quarantine** (for example, in the event of filter false positive) and messages with notifications on system operation may not reach their destination.



User Interface

When navigating within **Dr.Web Console for UNIX mail servers** sections, you cannot open the previous page using the standard **Back** function. If you click **Back** or use the corresponding key combination, the previous section of the main menu opens.

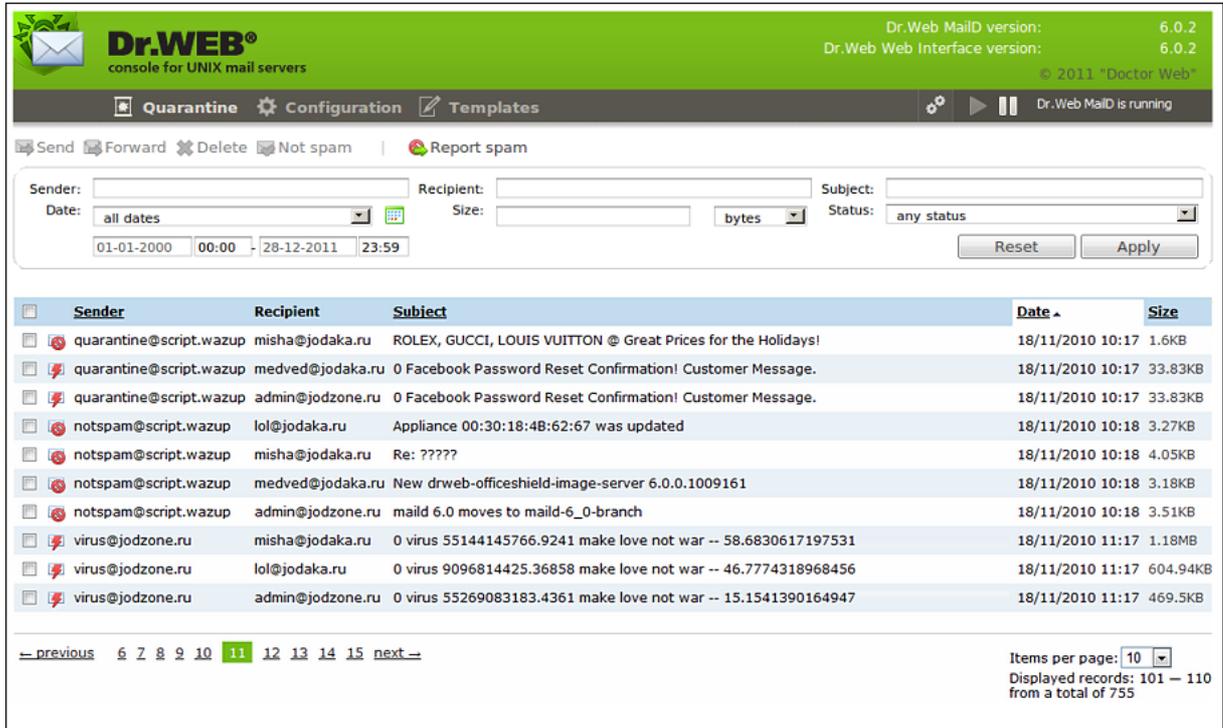


Figure 28. Dr.Web console for UNIX Mail Servers

Next to the module header, information about current versions of **Dr.Web MailD** and **Dr.Web for UNIX mail servers** web interface displays.

Under the module header, you can see the following three sections: **Quarantine**, **Configuration** and **Templates**. By default, the main page of the **Quarantine** section opens.

Next to the section headers you can see the following buttons: **Interface configuration** , **Start Dr.Web MailD**  and **Stop Dr.Web MailD** , as well as the current **Dr.Web MailD** status. When operating in the central protection mode, clicking **Stop Dr.Web MailD** stops all local **Dr.Web for UNIX mail servers** services running in this mode.



If **Dr.Web MailD** is operating in the [central protection mode](#), after access permissions to **Dr.Web ESS Control Center** settings are changed, reload the web interface page for the changes to take effect.

Quarantine

Messages filtered by **Dr.Web for UNIX mail servers** anti-virus or anti-spam module and considered to be malicious or spam, can be moved to **Quarantine**. On the **Quarantine** tab of the **Dr.Web for UNIX mail servers** web interface, all necessary tools for successful management of quarantined messages are collected.



Figure 29. Quarantine tab

The **Quarantine** tab contains the following elements:

- [toolbar](#)
- [filter panel](#)
- table with the [list of quarantined messages](#)
- list of additional [navigation tools](#) and its display settings.



Note that **Dr.Web Console for UNIX mail servers** does not allow management of "stalled" messages that reside in the /out/failed directory (for details on "stalled" messages, refer to the [description](#) of the **Sender** parameters).

For detection of such messages, review the directory. If it contains such messages, you can send it to its recipients by the means of `drweb-inject` [utility](#) or delete them using standard OS tools.

Toolbar

Toolbar items (except for the **Report spam** button) become active when a quarantined message is selected from the list.

Total messages selected: 2

<input type="checkbox"/>	St...	Sender	Recipient	Subject	Date	Size
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	With a big stick you will be the king of the beach.	28/04/10 12:27	2KB
<input checked="" type="checkbox"/>		yo-yo@anonhost	test@maildesk	Give us a call to get a diploma.	28/04/10 12:27	1KB
<input checked="" type="checkbox"/>		yo-yo@anonhost	test@maildesk	Одиночество закончилось	28/04/10 12:27	982b
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Предложение по рекламе	28/04/10 12:27	84KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	15% акций ЗАО Группы предприятий ОСТ продам	28/04/10 12:27	2KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Документы для прекращения договора	28/04/10 12:27	4KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	блоки	28/04/10 12:27	2KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Doctorate degree can be yours.	28/04/10 12:27	1KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Номенклатура деп отдела кадров	28/04/10 12:27	4KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Oprah certified wieght loss solution Acai Berri	28/04/10 12:27	2KB

**Figure 30. Toolbar**

Using the toolbar, you can:

- Send quarantined messages to their original recipients. Select required messages from the list and click  **Send**.
- Forward quarantined messages to an email address. Select required messages from the list and click  **Forward**. As a result, a window that contains the following fields opens: **Recipient** (email address of the recipient), **Subject** (subject of the message), **Message** (description of the forwarded message), **Attachments** (messages forwarded as attachments).
- Delete one or several quarantined messages. Select required messages from the list and click  **Delete** or press the DEL key on the keyboard.
- Notify our developers about false spam detection. Select messages which were filtered by anti-spam plug-in and moved to **Quarantine** by mistake and click  **Not Spam**. Once the button is clicked, a special message notifying on false detection with the attached text of the selected messages is automatically created and sent to vrnospam@drweb.com. The selected messages themselves are neither deleted from **Quarantine** nor sent to their recipients. If you still want to send these messages to recipients or forward them, use the corresponding buttons on the toolbar.
- Report spam. After you click  **Report spam**, a window opens where you can upload the file which contains the suspicious message and send it to **Doctor Web** laboratory.



This option cannot be applied to messages from the list. If a user considers a message that was not detected to be spam, it is required to save the message as a file to the file system.

Filter Panel

Filter panel simplifies processing of quarantined messages.

**Figure 31. Filter panel**

Using the filter system, you can select messages that satisfy the following criteria:

- **Sender** – email address of a message sender. In this field, you can enter a full email address or only a part of it.
- **Recipient** – email address of a message recipient. In this field, you can enter a full email address or only a part of it.
- **Subject** – any text to be looked for in the corresponding field of quarantined messages. As a result, only those messages are displayed, which **Subject** field contains the specified text string (both complete and partial match are allowed).
- **Date** – date when a message was moved to **Quarantine**. You can select a certain time period from the drop-down list or set the period using the calendar which opens after you click the  button. The following date options are available:
 - **all dates** – select all the messages stored in **Quarantine**.
 - **today** – select messages moved to **Quarantine** between the midnight and the present moment.
 - **yesterday** – select messages moved to **Quarantine** between the midnight of the previous day and the midnight of the current day.
 - **this week** – select all messages moved to **Quarantine** for the current week.



- o **this month** – select all messages moved to **Quarantine** for the current month.
- o **custom period** – select all messages for an arbitrary time period. You can use the calendar to specify boundaries of the required period.



Figure 32. Calendar

Calendar window opens automatically after you select the **custom period** option from the drop-down menu or when you click the  button. Specify the boundaries of the required time period and click **OK**. After the calendar window closes, the selected boundary values will appear in the corresponding fields. You can also specify the exact time value or time interval for the message search.



If you specify the time interval, only those messages will be selected, which were moved to **Quarantine** during the specified time period (including the boundaries). So, if you want to select messages received at a certain time, specify the same time values as period boundaries in the corresponding fields.

- **Size** – numerical value. By default, it is treated as a message size in bytes, but you can set another unit (KB or MB) using the corresponding drop-down list. If this criterion is used, only those messages are selected which size is greater than or equal to the specified value. If the **Size** value is set to 0, this criterion is ignored when searching.
- **Status** – reason why the message was moved to **Quarantine**. The following statuses are available for filtering:
 - o **virus** – message was considered infected by the virus and moved to **Quarantine** by **Dr.Web for UNIX mail servers** anti-virus module
 - o **spam** – message was considered spam and moved to **Quarantine** by the **Dr.Web for UNIX mail servers** anti-spam module
 - o **rule** – message was sent to **Quarantine** according to the internal message processing rules
 - o **processing error** – an error emerged during processing of this message, so it was moved to **Quarantine**

After all selection criteria are specified, click **Apply**. To return to the defaults, click **Reset**.

List of Messages

If the **Quarantine** folder contains messages, their list displays in tabular form on the **Quarantine** tab.



Total messages selected: 2

<input type="checkbox"/>	St...	Sender	Recipient	Subject	Date	Size
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	With a big stick you will be the king of the beach.	28/04/10 12:27	2KB
<input checked="" type="checkbox"/>		yo-yo@anonhost	test@maildesk	Give us a call to get a diploma.	28/04/10 12:27	1KB
<input checked="" type="checkbox"/>		yo-yo@anonhost	test@maildesk	Одиночество закончилось	28/04/10 12:27	982b
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Предложение по рекламе	28/04/10 12:27	84KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	15% акций ЗАО Группы предприятий ОСТ продам	28/04/10 12:27	2KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Документы для прекращения договора	28/04/10 12:27	4KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	блоки	28/04/10 12:27	2KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Doctorate degree can be yours.	28/04/10 12:27	1KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Номенклатура деп отдела кадров	28/04/10 12:27	4KB
<input type="checkbox"/>		yo-yo@anonhost	test@maildesk	Oprah certified wieght loss solution Acai Berri	28/04/10 12:27	2KB

Figure 33. List of messages

Administrator has access to all quarantined messages sent to recipients from groups controlled and managed by this Administrator.

Tabular data is organized in the following way:

- **Status** column – contains information about the message status (reason why it was moved to **Quarantine**). Status is displayed as one of the following icons:



– message contains virus



– message is marked as spam



– message is quarantined according to internal filtering rules



– message evoked an error during processing.

When you move the pointer over the status icon, a tooltip appears with detailed description of the reason why the message was quarantined.

- **Sender** column – contains information about the sender's email address. You can sort messages by the field in direct and reverse alphabetical order.
- **Recipient** column – contains information about the recipient's email address. You can sort messages by the field in direct and reverse alphabetical order.
- **Subject** column – contains information about the message subject. You can sort messages by the field in direct and reverse alphabetical order.
- **Date** column – contains information about the date when the message was moved to **Quarantine**. For messages quarantined in the last twenty four hours only time is specified. You can sort messages by date in ascending and descending order.
- **Size** column – contains information about message size. You can sort messages by size in ascending and descending order.

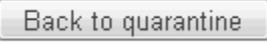
To select a message from the list, set the corresponding check box. To select all messages in the list, select the check box in the left corner of the table header

Values of the **Recipient**, **Subject** and **Date** fields are the links to the corresponding message: clicking any of them switches you to the message.



Figure 34. Quarantined message

On this screen, you can view message content, source (by clicking the  **View source** link), headers (be clicking the  **Headers** link) and attachments (if there are any).

Click the  button to return to the main **Quarantine** screen.

Navigation Pane



Figure 35. Navigation pane

Additional navigation tools include:

- page navigator to go to the previous or next page of the table; displayed as **previous** and **next** links (you can use the following keyboard shortcut: press CTRL together with the right or left arrow to go to the next or previous page correspondingly);
- page navigator for browsing through all pages of the table; displayed as page number links. Link to the current page is not active and is marked with green colour.
- indicator of the total number of messages in the list and number of messages displayed on the page.

You can adjust the number of messages displayed on the page. To do that, select one of the available numbers from the drop-down list: 10, 20, 50, 100. After you select the required number, the table is automatically reformatted.



After the table is reformatted or its content is sorted, all previous selections are removed.

Configuration

You can select required parameter values from drop-down lists by clicking  buttons or specifying the values manually in the corresponding text fields. Detailed description of each parameter is provided in the Help system. To use it, click **more >>**. After changing any parameter value, you can immediately undo the change by clicking  or restore the default value by clicking . The latter is always available, even after changes are saved.



To revise all changes, click **Preview**. On the open page, you can select the changes that you want to save by selecting the corresponding check box in the **Save** column.

Parameter	Old value	New value	Save
Use secure hash	no	yes	<input checked="" type="checkbox"/>
Accept control messages only from trusted networks	yes	no	<input checked="" type="checkbox"/>
License limint action	pass	pass,quarantine,notify	<input checked="" type="checkbox"/>
Processing errors action	pass	pass,quarantine	<input checked="" type="checkbox"/>
Scan encoded headers (headersfilter)	yes	no	<input checked="" type="checkbox"/>
Before queue plug-ins	vaderetro,modifier	vaderetro,modifier,headersfilter	<input checked="" type="checkbox"/>
Hostname (Firebird)	localhost	host	<input checked="" type="checkbox"/>
Response size limit (ODBC)	0	4	<input checked="" type="checkbox"/>

Figure 36. Preview screen

- If you want to discard the changes, click **Cancel**.
- If you want to make additional changes, click **Continue Editing** to return to the previous page.
- Click **Save** to save the changes or **Apply and Save** to save and apply them immediately.



Basic Settings Tab

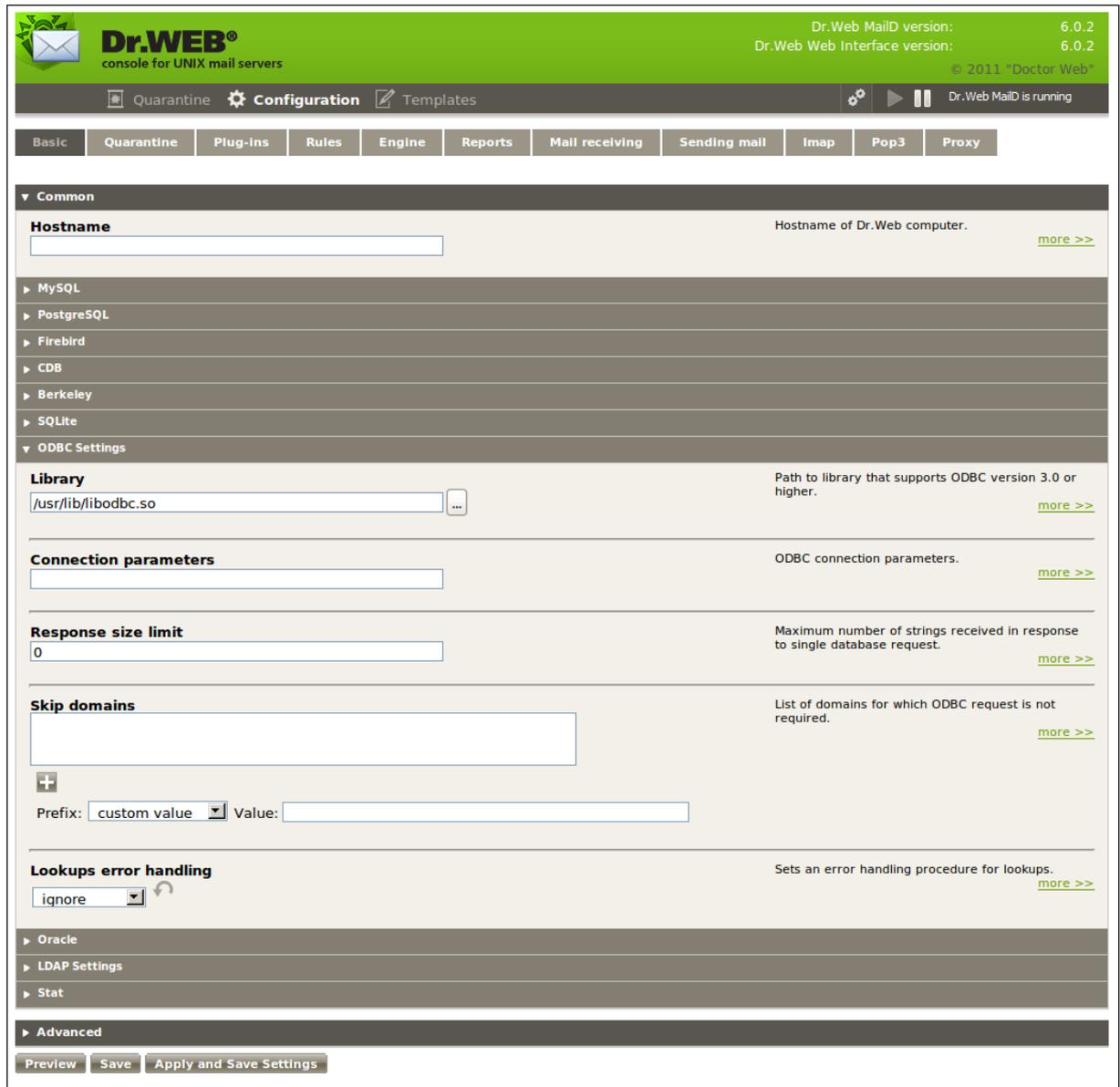


Figure 37. Basic settings

On this tab, you can configure export of statistics as well as interaction between **Dr.Web MailD** and various databases. You can select parameter values from the drop-down menus or specify them manually in the corresponding text fields. Queries to the LDAP server must start with double or triple slash.

Example:

```
//127.0.0.1/dc=origin?description?sub?(cn=$u)
```

Double slash is used when you need to specify the address of the LDAP server.

Example:

```
///?description?sub?(cn=$u)
```

Triple slash instructs to use the server specified as the **Hostname** parameter value in the [LDAP]



section of the **Dr.Web MailD** configuration file.

Quarantine Tab

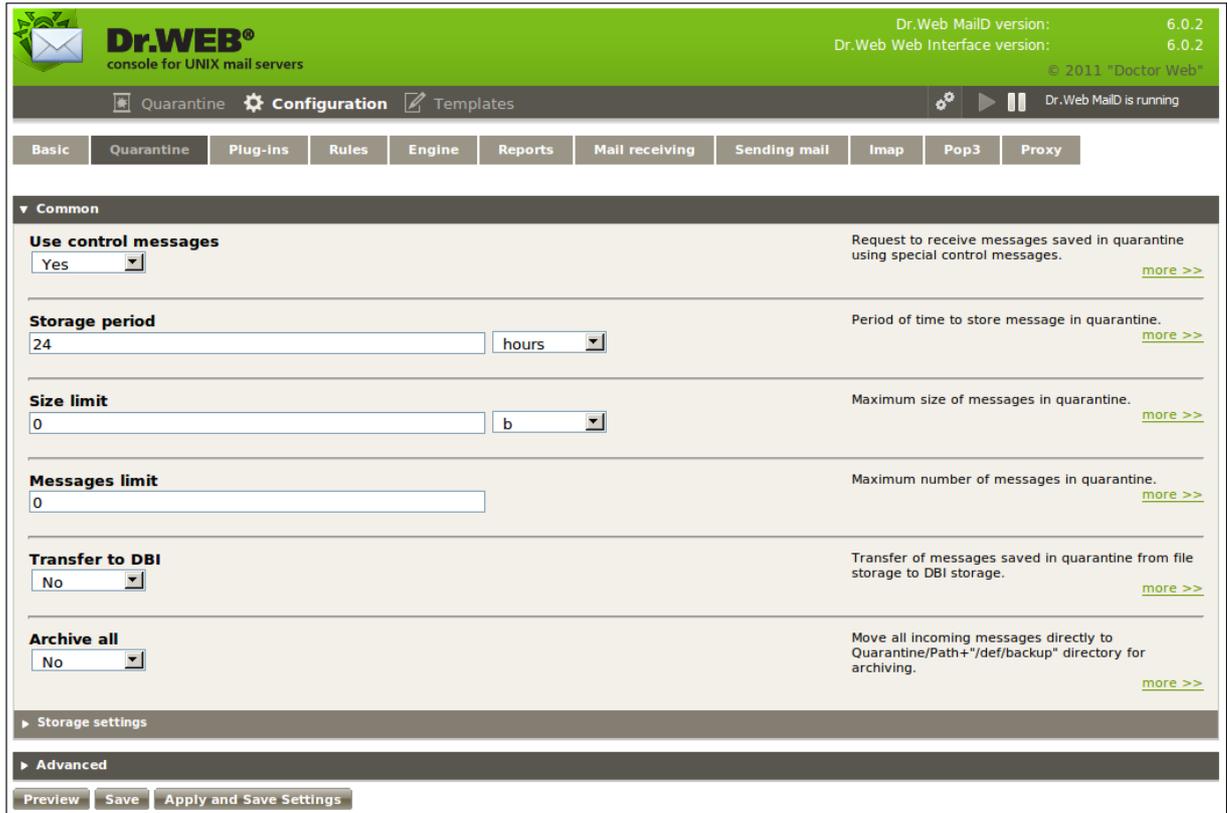


Figure 38. Quarantine settings

On this tab, you can configure main **Quarantine** settings: the period of time to store messages in **Quarantine**, privileges to access quarantined messages, renaming rules, interaction with **DBI** storage.

Plug-ins Tab

This tab contains general settings of all **plug-ins** included in **Dr.Web MailD**. To adjust settings of a certain plug-in, open the corresponding tab.



The screenshot shows the Dr.Web MailD configuration console. At the top, it displays the version (6.0.2) and copyright information. The main navigation bar includes 'Quarantine', 'Configuration', and 'Templates'. Below this, a sub-menu highlights 'Anti-spam', 'Headers filter', 'Anti-virus', and 'Modifier'. The 'Common' section is expanded to show plug-in settings. Under 'Before queue plug-ins', 'antispam' and 'modifier' are active, while 'headersfilter' and 'antivirus' are available to be added. Under 'After queue plug-ins', 'antivirus' is active, and 'antispam', 'headersfilter', and 'modifier' are available. There are also input fields for message size limits for both before and after queue plug-ins, set to 0 with a unit dropdown set to 'b'. At the bottom, there are buttons for 'Preview', 'Save', and 'Apply and Save Settings'.

Figure 39. Settings of plug-ins

Values of additional actions such as **redirect**, **add-header** and **add-score** are not separated by parentheses "(" and ")"; additional actions are set as immediate values:

- for **redirect** action a list of address separated with "|" is specified:
address1@domain | address2@domain | address3@domain;
- for **add score** action, only the score value is specified;
- **add header** value is specified as [NAME:]BODY, where NAME is the header name (X-DrWeb-MailD by default), and BODY is the header value.

Value of **add score** additional action must be escaped with double quotation marks when added to the configuration file (for details, see [Allowed Actions](#)).

For example, to add the word "Infected" as a header for infected files on the **Anti-virus** tab, the following string must be added to the plug-in configuration file:

```
Infected = cure, quarantine, notify, "add-header(escaped!)"
```

Anti-Spam Tab

This tab contains general settings of **Vaderetro anti-spam plug-in** included in **Dr.Web for UNIX mail servers**.



The screenshot shows the Dr.Web MailD configuration interface. At the top, there is a green header with the Dr.Web logo and version information (Dr.Web MailD version: 6.0.2, Dr.Web Web Interface version: 6.0.2). Below the header is a navigation bar with tabs for 'Quarantine', 'Configuration', and 'Templates'. The 'Configuration' tab is active, and a sub-menu is open showing 'Anti-spam', 'Headers filter', 'Anti-virus', and 'Modifier'. The 'Anti-spam' sub-tab is selected. The main content area is divided into several sections:

- Common**: Contains settings for 'Full check' (Yes), 'Ignore embedded domains' (Yes), 'Add version header' (No), and 'Add spam status header' (No).
- Unconditional spam action**: Contains 'Main action' (pass) and 'Additional actions' (quarantine, redirect, add header).
- Spam action**: Contains 'Main action' (pass) and 'Additional actions' (quarantine, redirect, add header).
- Black List**: Contains a 'Prefix' (custom value) and a 'Value' field.
- Size limit**: Contains a 'Size limit' field (0) and a unit dropdown (b).
- Log verbosity level**: Contains a dropdown menu (info).

At the bottom of the interface, there are buttons for 'Preview', 'Save', and 'Apply and Save Settings'.

Figure 40. Anti-spam settings

Headers Filter Tab

This tab contains general settings of [Dr.Web Headersfilter plug-in](#) which allows filtering of messages by their headers.



The screenshot displays the 'Configuration' page for 'Header filter' settings in the Dr.Web console. The interface is organized into several sections:

- Common:**
 - Scan encoded headers:** A dropdown menu is set to 'Yes'. A note states 'Headers scan before decoding.' with a 'more >>' link.
 - Reject:** A large text input field for a condition. Below it, a '+', a 'Prefix' dropdown (set to 'custom value'), and a 'Value' text input field. A note mentions 'Message filtering rules.' with a 'more >>' link.
 - Reject embedded parts:** A large text input field for a condition. Below it, a '+', a 'Prefix' dropdown (set to 'custom value'), and a 'Value' text input field. A note states 'Rules are similar to those from RejectCondition parameter, but they affect only headers of attached objects.' with a 'more >>' link.
 - Accept embedded parts:** A large text input field for a condition. Below it, a '+', a 'Prefix' dropdown (set to 'custom value'), and a 'Value' text input field. A note states 'Rules are similar to those from AcceptCondition parameter, but they affect only headers of attached objects.' with a 'more >>' link.
- Action:**
 - Main action:** A dropdown menu set to 'reject'. A note states 'Action to be applied to filtered messages.' with a 'more >>' link.
 - Additional actions:** A list of actions: 'notify' (with a close 'x' icon), 'quarantine', 'redirect' (with a text input field), 'add header' (with a text input field), and 'add score' (with a text input field).
- Use custom reply:** A dropdown menu set to 'No'. A note states 'Reply strings to be used as SMTP reply when messages have been rejected.'
- Log verbosity level:** A dropdown menu set to 'info'. A note states 'Plug-in log verbosity level.'

At the bottom, there is an 'Advanced' section header and three buttons: 'Preview', 'Save', and 'Apply and Save Settings'.

Figure 41. Header filter settings

The string "HEADER = regular_expression" must be specified in the **Value** field for all ~Condition parameters. In the **redirect** text field of the **Action** parameter section you can specify an email address where filtered messages are to be redirected.



Anti-Virus Tab

This tab contains general settings of **Drweb anti-virus-plug-in** included in **Dr.Web for UNIX mail servers**.

The screenshot displays the Dr.Web console interface for configuring the anti-virus plug-in. The top navigation bar includes 'Quarantine', 'Configuration', and 'Templates'. Below this, a series of tabs are visible: 'Basic', 'Quarantine', 'Plug-Ins', 'Rules', 'Engine', 'Reports', 'Mail receiving', 'Sending mail', 'Imap', 'Pop3', and 'Proxy'. The 'Anti-virus' tab is currently selected, and within it, the 'Common' sub-tab is active.

The configuration is organized into several sections:

- Common:** Includes 'Socket address' (set to 'pid:/var/drweb/run/drwebd.pid'), 'Timeout' (set to '30 seconds'), and 'Heuristic analysis' (set to 'Yes').
- Infected:** Includes 'Main action' (set to 'cure') and 'Additional actions' (including 'quarantine' and 'notify').
- Processing errors:** Includes 'Main action' (set to 'reject') and 'Additional actions' (including 'quarantine' and 'notify').
- Size limit:** Includes 'Size limit' (set to '0') and 'Log verbosity level' (set to 'info').

At the bottom of the configuration area, there are buttons for 'Preview', 'Save', and 'Apply and Save Settings'.

Figure 42. Anti-virus settings

In the **redirect** text field of any parameter that manages actions on messages, you can specify an email address where filtered messages are redirected (by default, an email address specified as the **RedirectMail** parameter value on the **Engine** tab is used).



Advanced settings allow to configure custom replies sent to users upon message block.

Use custom reply No	Reply strings to be used as SMTP reply when messages have been rejected.
Use TCP_NODELAY No	Enable TCP_NODELAY parameter for socket. more >>
Log file limit 50 KB	Maximum size of drwebd log file. more >>
Infected reply "DrWEB Antivirus: Message is rejected because it contains"	Reply string to be used as SMTP reply when Infected = reject or Incurable = reject actions are applied, and also when UseCustomReply = yes. more >>
Malware reply "DrWEB Antivirus: Message is rejected because it contains"	Reply string to be used as SMTP reply when Adware, Dialers, Jokes, Riskware, Hacktools = reject actions are applied, and also when UseCustomReply = yes. more >>
Suspicious reply "DrWEB Antivirus: Message is rejected because it contains"	Reply string to be used as SMTP reply when Suspicious = reject action is applied, and also when UseCustomReply = yes. more >>
Error reply "DrWEB Antivirus: Message is rejected due to software err"	Reply string to be used as SMTP reply when ScanningErrors, ProcessingErrors = reject actions are applied, and also when UseCustomReply = yes. more >>
Block by filename reply "DrWEB MailD: Message is rejected due to filename patter"	Reply string to be used as SMTP reply when BlockByFilename = reject action is applied, and also when UseCustomReply = yes. more >>
IPC level alert	IPC library log verbosity level.
Syslog facility Mail	Syslog facility type generating notifications on Dr.Web events if syslogd is used for Dr.Web and its components activity logging. more >>
Libraries []	Path to plug-in libraries. more >>
Section []	Name of section of the configuration file, where parameters regulating plug-in operation can be found.

Preview Save Apply and Save Settings

Figure 43. Anti-virus advanced settings

Modifier Tab

This tab contains general settings of **Dr.Web Modifier plug-in** included in **Dr.Web for UNIX mail servers**.



Rules Tab

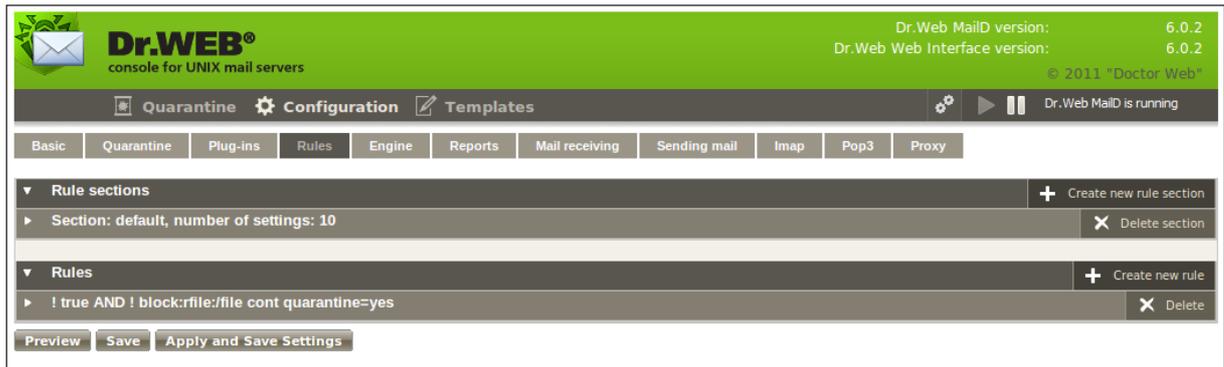


Figure 45. Message processing rules

This tab contains settings of the [Rules] section of Dr.Web MailD configuration file. You can specify separate Rules as well as rule sets. To create a separate rule, click **Create new rule button**. To edit a rule (either newly created or an old one), click it.

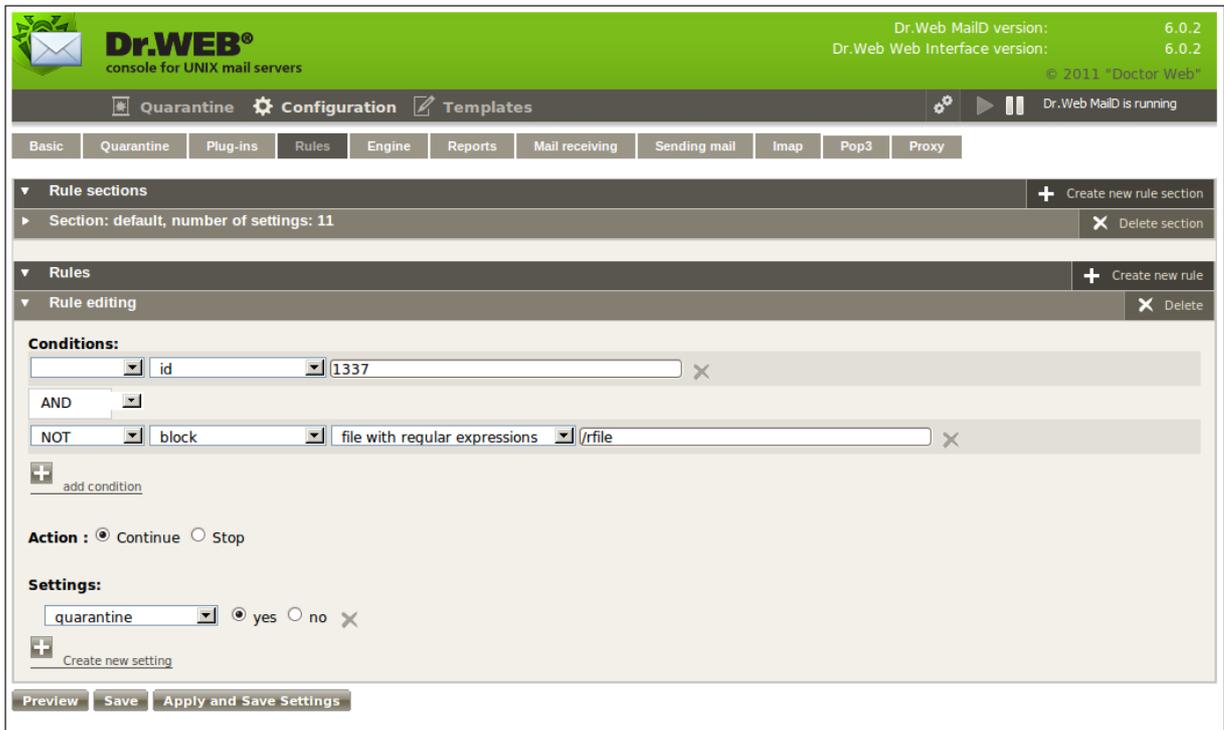


Figure 46. Rule editing

When editing a rule, specify parameters in all of the following sections: **Conditions**, **Actions** and **Settings**. Conditions can contain logical operators.



Please note that the rule editor, included in the current version of **Dr.Web Console for UNIX mail servers**, has the following restrictions:

1. complex rules containing associations are not supported.
2. rules that override the **SkipDomains** parameter value for a **Lookup** are not supported.

Such rules can be edited only in the **configuration file** or via web interface of **Dr.Web ESS Control Center** (in the enterprise mode). However, such rules are not displayed correctly in the Rule editor included in **Dr.Web Console for UNIX mail servers**.



Please note that local filtering rules of **Dr.Web Modifier plug-in** must be specified according to the same principle as the `modifier/LocalRules` parameter value in the [configuration file](#), that is, escape quotation marks and slash symbols, if any:

- single quote (') is not escaped;
- double quote (") is escaped with 6 back slashes;
- back slash is escaped with 7 back slashes.

The picture below illustrates an example of a local filtering rule `modifier/LocalRules` that triggers if a message subject is equal to the following string: `It's "big\small" text`.

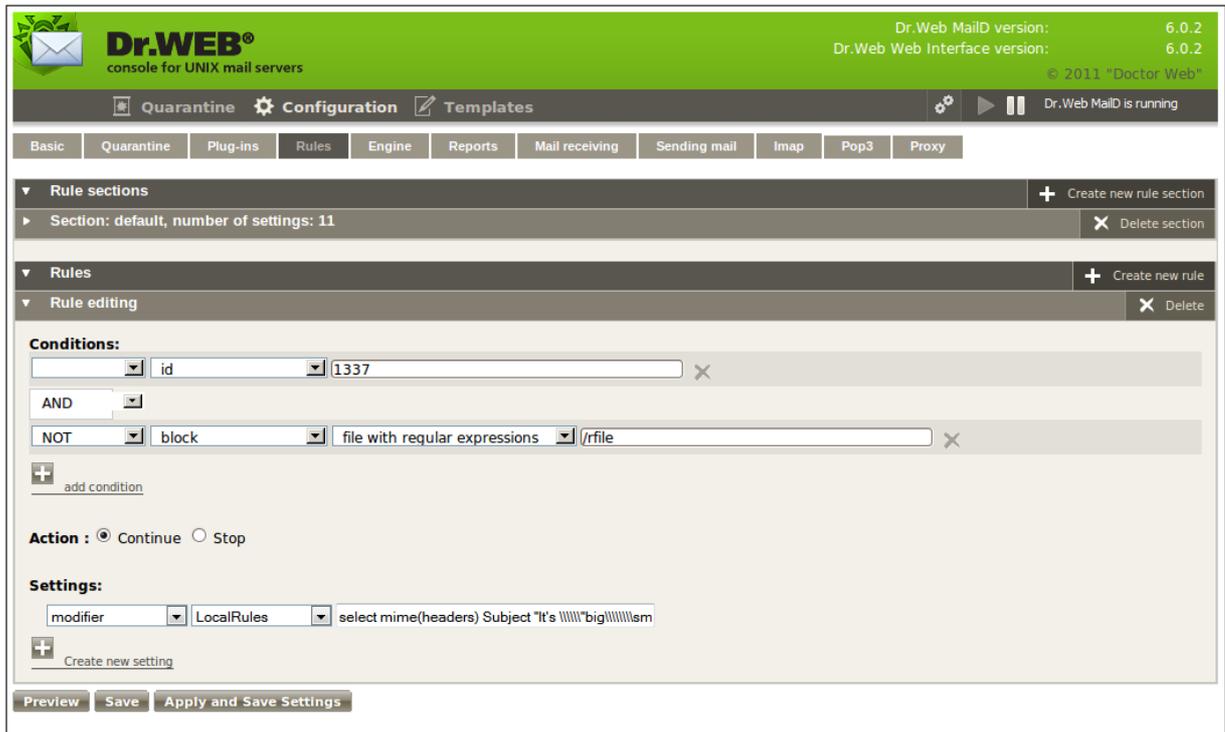


Figure 47. Rule editing

Rule sets are created by clicking **Create new rule section**. To edit a rule set (either newly created or an old one), click it.



The screenshot shows the Dr.Web MailD configuration interface. At the top, there is a green header with the Dr.WEB logo and version information: "Dr.Web MailD version: 6.0.2" and "Dr.Web Web Interface version: 6.0.2". Below the header is a navigation bar with tabs for "Quarantine", "Configuration", and "Templates". The "Configuration" tab is active, and a sub-menu is open showing "Basic", "Quarantine", "Plug-ins", "Rules", "Engine", "Reports", "Mail receiving", "Sending mail", "Imap", "Pop3", and "Proxy". The "Rules" sub-tab is selected, and the "Editing rule section [default]" window is open. This window contains a "Settings:" section with a list of rule settings, each with a "Notify" dropdown, a category dropdown, and a value field. The settings are: Notify (block), Notify (Virus, allow(any)), Notify (Cured, allow(admin:sender)), Notify (Skip, block), Notify (Archive, allow(admin)), Notify (Error, allow(admin)), Notify (Rule, allow(admin)), Notify (License, allow(admin)), and Notify (Malware, allow(any)). There is also a "html" setting with radio buttons for "yes" (selected) and "no". At the bottom of the "Settings" section is a "Create new setting" button. Below the settings is a "Rules" section with a "Create new rule" button and a rule entry: "! true AND ! block:file:/file cont quarantine=yes". A "Delete" button is next to the rule entry. At the bottom of the interface are buttons for "Preview", "Save", and "Apply and Save Settings".

Figure 48. Editing rule section



Engine Tab

Dr.WEB®
console for UNIX mail servers

Dr.Web MailD version: 6.0.2
Dr.Web Web Interface version: 6.0.2
© 2011 "Doctor Web"

Quarantine Configuration Templates

Basic Quarantine Plug-Ins Rules **Engine** Reports Mail receiving Sending mail Imap Pop3 Proxy

Common

Protected networks List of protected networks. [more >>](#)
127.0.0.0/8 ✕
+
Prefix: custom value Value:

Protected domains List of protected domains.
+
Prefix: custom value Value:

Include subdomains Include subdomains in protected domains list.
Yes

Redirect to E-mail address to send messages to when Redirect action is used.
root@localhost

Processing errors action Action applied to messages invoked scanning errors.
Main action: pass
Additional actions:
+ quarantine
+ redirect
+ notify
+ add header
+ add score

Maximum score Maximum message score. [more >>](#)
10000

Action for maximum score Actions applied to the message when its score exceeds the threshold value specified in MaxScore parameter. [more >>](#)
Main action: pass
Additional actions:
+ quarantine
+ redirect
+ add header
+ add score

Advanced

Preview Save Apply and Save Settings

Figure 49. General Engine settings

On this tab, you can specify an email address where messages filtered by a plug-in are to be redirected.



To do this, enter the required address in the corresponding `redirect` filed. You can also enable management of `drweb-maild` by the means of [control email messages](#).

To configure [pool options](#) and custom replies, specify required values in the corresponding fields.

Advanced

Input pool options Thread pool settings for processing before queue.
Current values:
 auto
 minimum
 minimum maximum
timeout
stack_size
loglevel
stat

Pid file Path to pid-file of drweb-maild process.

Use custom reply Custom messages in SMTP sessions. [more >>](#)

Get IP from header Use "Received" header value as Client IP address if it is not identified by Receiver component.

Skip DSN on blocking Whether to send DSN report, when program fails to pass return code to Receiver component after performing Reject or Tempfail actions.

Mime parts limit Maximum number of MIME parts in a message. [more >>](#)

Nested mime parts limit Maximum number of nested MIME parts in the message. [more >>](#)

▶ MailBase settings

Preview Save Apply and Save Settings

Figure 50. Advanced Engine settings

In the **MailBase** section, you can configure mail database settings.



MailBase settings

Backup database	<input type="text" value="/var/drweb/msqs/db/maildb.backup"/> ...	Mail database backup file name. more >>
Backup period	<input type="text" value="0"/> seconds	Time period for database backup. more >>
Deletion period	<input type="text" value="48"/> hours	Maximum period of time for message to be stored in mail database. more >>
Additional timeout	<input type="text" value="2"/> hours	Additional time for message processing. more >>
Body size limit	<input type="text" value="1"/> KB	Maximum size of message stored in mail database. more >>
Storage size	<input type="text" value="0"/> b	Maximum mail database size in bytes. more >>
Pool size	<input type="text" value="0"/> b	Maximum mail database pool size. more >>
Messages storage limit	<input type="text" value="100000"/>	Maximum number of messages stored in mail database. more >>
Send timeout	<input type="text" value="30"/> seconds	Timeout for plugin to perform an asynchronous check of a message. more >>

Figure 51. MailBase settings



Reports Tab

The screenshot displays the 'Reports' configuration tab in the Dr.Web console. The top navigation bar includes 'Quarantine', 'Configuration', and 'Templates'. The main content area is divided into 'Common' and 'Advanced' sections. The 'Common' section contains the following settings:

- Send reports:** A dropdown menu set to 'Yes'. Description: Report sending.
- Reports schedule:** A text input field containing '00:00:00'. Description: Report schedule. [more >>](#)
- E-mail address:** An empty text input field. Description: E-mail address(es) to send reports to. [more >>](#)
- Plug-in names:** An empty text input field. Description: List of plug-ins report is created for. [more >>](#)
- Top list size:** A text input field containing '20'. Description: Display in a report the lists of frequently blocked objects and addresses from which the maximum number of blocked objects has been sent. [more >>](#)
- Statistics storage period:** A text input field containing '31' and a dropdown menu set to 'days'. Description: Maximum period of time to store statistics in reports database. [more >>](#)
- Administrator e-mail:** A text input field containing 'root@localhost'. Description: Administrator e-mail address. [more >>](#)
- Filter e-mail:** A text input field containing 'root@localhost' with a refresh icon. Description: E-mail address specified in "From" header of messages with reports.
- Notification languages:** A multi-select field with 'en' selected and a refresh icon. Description: Language(s) used in reports generation. Below the field are buttons for '+ ja' and '+ ru'.

The 'Advanced' section is currently collapsed. At the bottom of the page, there are three buttons: 'Preview', 'Save', and 'Apply and Save Settings'.

Figure 52. Reports settings

On this tab, you can configure statistic report options, for example, set the schedule for sending reports to the Administrator or period of report storage in the database.



Mail Receiving Tab

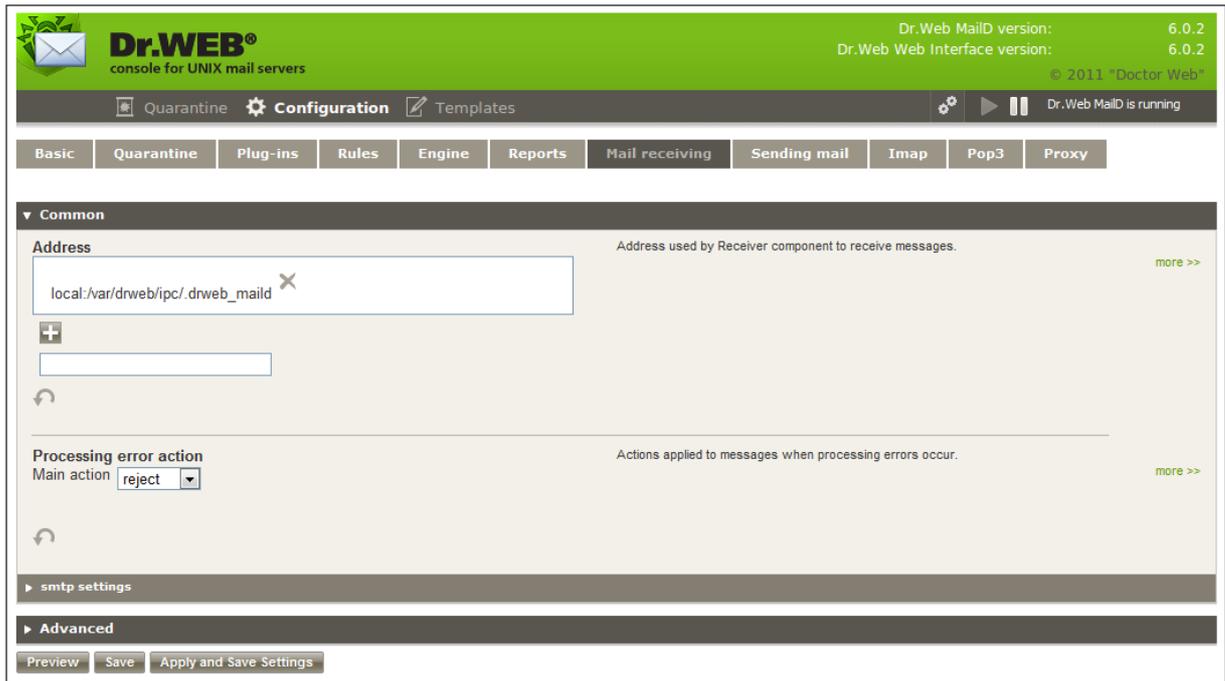


Figure 53. General mail receiving settings

On this tab, you can specify one or several addresses for receiving SMTP/LMTP requests and actions to be applied to messages which caused processing errors.

On the tab with advanced settings, you can configure interaction between **Dr.Web MailD** and the used MTA.



Advanced

Pool settings Thread pool settings.
Current values:

auto
 minimum
 minimum maximum

timeout
stack_size
loglevel
stat

Direct connections with clients Accept connections directly from clients.
 [more >>](#)

Stalled messages processing timeout Timeout to process stalled messages.
 [more >>](#)

Command timeout Timeout to execute single command.

Message timeout Timeout to receive single message.

Add Received header Add "Received" header to all received messages.

Return on reject Receiver component policy in case of Reject action.
 [more >>](#)

Figure 54. Advanced mail receiving settings



Current version of **Dr.Web for UNIX mail servers** does not support configuration of simultaneous usage of several **Receiver/Sender** components via the web interface.



Sending Mail Tab

Dr.WEB®
console for UNIX mail servers

Dr.Web MailD version: 6.0.2
Dr.Web Web Interface version: 6.0.2
© 2011 "Doctor Web"

Quarantine Configuration Templates Dr.Web MailD is running

Basic Quarantine Plug-ins Rules Engine Reports Mail receiving **Sending mail** Imap Pop3 Proxy

Common

Use secure hash Add "SecureHash" header to all outgoing messages. [more >>](#)

Secure hash "SecureHash" header contents. [more >>](#)
!!!----- EDIT THIS !!!

Advanced

Pool settings Thread pool settings.
Current values:
 auto
 minimum
 minimum maximum
timeout seconds
stack_size GB
loglevel
stat

Submit directory Directory where drweb-cpp-sender module submits messages that will be sent by CommuniGate Pro MTA.

Submit filenames mode Naming convention for file names of submitted messages. [more >>](#)

Submit filenames prefix Prefix for file names of submitted messages. [more >>](#)

Submit file permissions Permissions for created notifications or cured messages.

	Read	Write	Execute	SUID bit	
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Others	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Access privileges 0600

Preview Save Apply and Save Settings

Figure 55. Mail sending settings

On this tab, you can specify actions to be applied to outgoing messages and set timeouts for command execution and message processing by [Dr.Web Daemon component](#) and [plug-ins](#).



Note that [Dr.Web Console for UNIX mail servers](#) does not allow management of "stalled" messages that reside in the /out/failed directory (for details on "stalled" messages, refer to the [description](#) of the **Sender** parameters).

For detection of such messages, review the directory. If it contains such messages, you can send it to its recipients by the means of `drweb-inject` [utility](#) or delete them using standard OS tools.



IMAP Tab

The screenshot shows the Dr.Web MailD configuration interface. At the top, it displays the Dr.Web logo and version information: Dr.Web MailD version: 6.0.2 and Dr.Web Web Interface version: 6.0.2. The navigation bar includes tabs for Quarantine, Configuration (selected), and Templates. Below the navigation bar, there are several tabs: Basic, Quarantine, Plug-ins, Rules, Engine, Reports, Mail receiving, Sending mail, Imap (selected), Pop3, and Proxy. The main content area is titled 'General' and contains several sections:

- Callback pool settings:** Settings of auxiliary thread pool. Current values: auto, minimum block, minimum allow(a) maximum allow(a). Other fields: timeout block days, stack_size allow(a) b, loglevel info, stat no.
- Listen address:** A list of socket addresses used to receive requests from clients. Field: inet:5200@0.0.0.0. Action: allow(admin).
- Client TLS settings:** TLS/SSL settings for client communications over IMAP.
- Server TLS settings:** TLS/SSL settings for server communications over IMAP.
- Server address:** Address used by the filter to connect to IMAP server. Field: inet:imap@127.0.0.1. Action: allow(admin).
- Filter errors action:** Action to be applied to a message, when some error emerges before the message is passed to the drweb-maild module. Main action: reject.
- Pool settings:** Settings of main thread pool. Current values: auto, minimum allow(a), minimum allow(a) maximum mail.co. Other fields: timeout, stack_size GB, loglevel quiet, stat no.

At the bottom, there is an 'Advanced' section and buttons for Preview, Save, and Apply and Save Settings.

Figure 56. IMAP settings

On this tab, you can specify IMAP filter settings to check mail received via the IMAP protocol.



POP3 Tab

Dr.WEB®
console for UNIX mail servers

Dr.Web MailD version: 6.0.2
Dr.Web Web Interface version: 6.0.2
© 2011 "Doctor Web"

Quarantine Configuration Templates Dr.Web MailD is running

Basic Quarantine Plug-Ins Rules Engine Reports Mail receiving Sending mail Imap **Pop3** Proxy

General

Settings of auxiliary pool Settings of auxiliary thread pool. [more >>](#)
Current values:
 auto
 minimum 10000
 minimum 7331 maximum block
timeout allow(a) days
stack_size allow(a) MB
loglevel alert
stat no

Listen address A list of socket addresses used to receive requests from clients. [more >>](#)
inet:5110@0.0.0.0
+
block

Client TLS Settings TLS/SSL settings for client communications over POP3. [more >>](#)

Server TLS settings TLS/SSL settings for server communications over POP3. [more >>](#)

Server address Address used by the POP3 filter to connect to POP3 server.
inet:pop3@127.0.0.1
+
allow(admin)

Filter error action Action to be applied to message, when some error emerges before the message is passed to the drweb-maild module. [more >>](#)
Main action reject

Pool settings Settings of main thread pool. [more >>](#)
Current values:
 auto
 minimum allow(a)
 minimum allow(a) maximum allow(a)
timeout allow(a) minutes
stack_size MB
loglevel
stat no

Advanced

Preview Save Apply and Save Settings

Figure 57. POP3 settings

On this tab, you can specify [POP3 filter](#) settings to [check mail received via the POP3 protocol](#).



Proxy Tab

Dr.Web MailD version: 6.0.2
Dr.Web Web Interface version: 6.0.2
© 2011 "Doctor Web"

Quarantine Configuration Templates Dr.Web MailD is running

Basic Quarantine Plug-Ins Rules Engine Reports Mail receiving Sending mail Imap Pop3 Proxy

Client

Address List of socket addresses used by the Sender component to receive requests from drweb-proxy-server components to send mail. [more >>](#)

inet:8066@0.0.0.0 X

Proxy servers List of socket addresses used by drweb-proxy-server components. [more >>](#)

inet:8088@SERVER-IP X

Main pool settings Settings of a thread pool processing requests from Receiver component. [more >>](#)

Current values:

auto
 minimum
 minimum 2 maximum 20

timeout seconds
stack_size b
loglevel quiet
stat no

Sender pool settings Settings of thread pool used for processing requests from drweb-proxy-server components to send mail via the Sender component. [more >>](#)

Current values:

auto
 minimum
 minimum 2 maximum 20

timeout minutes
stack_size b
loglevel quiet
stat yes

Server

Preview Save Apply and Save Settings

Figure 58. Proxy settings

On this tab, you can configure operation of the [internal proxy](#) which enables different **Dr.Web for UNIX mail servers** components residing on different hosts interact with each other.

Templates

This section contains templates of [MailD notifications](#) that are sent to various recipients upon detection of malicious objects in mail messages and occurrence of errors during [Dr.Web Daemon component](#) or [plug-ins](#) operation.

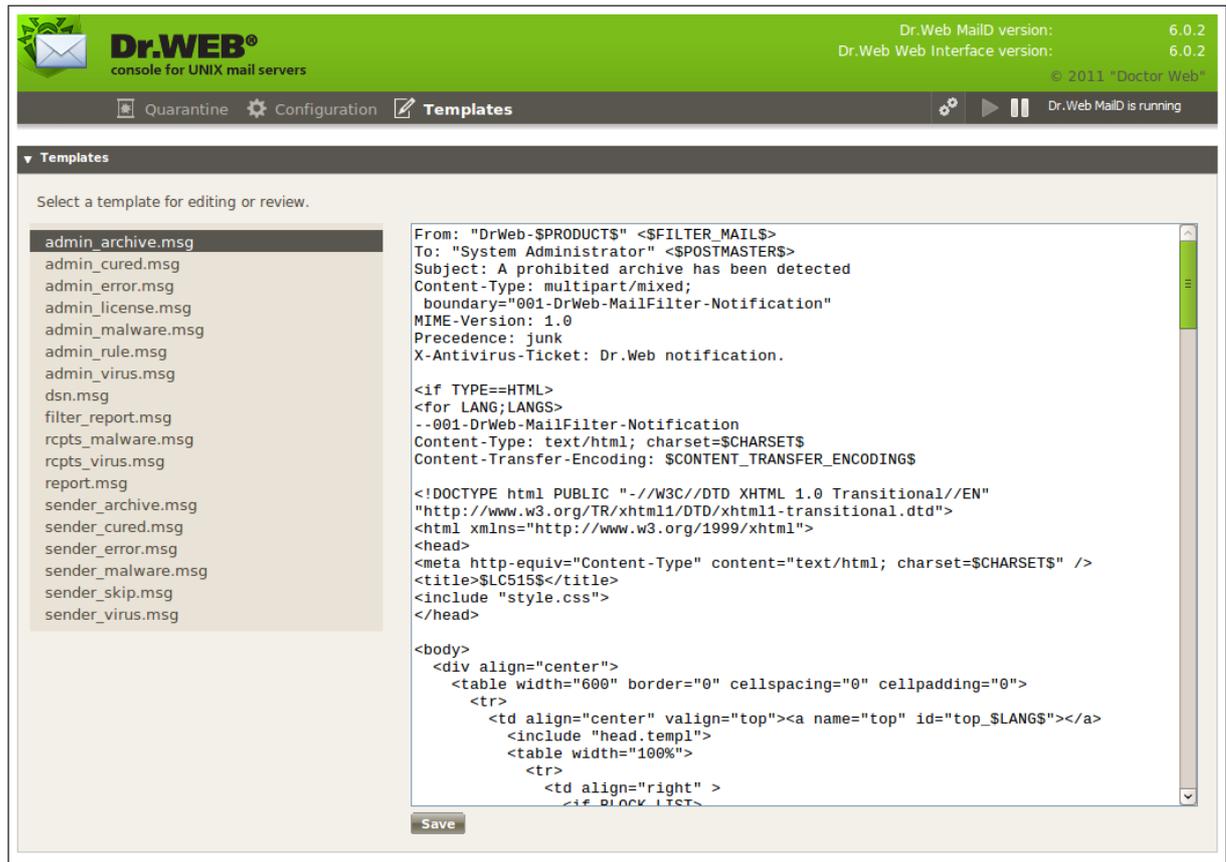


Figure 59. Templates

Running in Enterprise Mode

To start **Dr.Web Console for UNIX mail servers** in the central protection mode, configure **Dr.Web Agent** as described in the [corresponding section](#). After making necessary changes, click  on the navigational menu at the top of the page. In the open window, set Central Protection Mode parameter value to Yes or Auto.

Central Protection Mode parameter can have one of the following three values:

- No – in this mode **Dr.Web Console for UNIX mail servers** interacts with local configuration file and does not have access to the configuration received by **Dr.Web Agent** from **Dr.Web Enterprise Server**. Changes made to the configuration in this mode take effect only after **Dr.Web Agent** is set to operate in the Standalone mode.
- Yes – in this mode **Dr.Web Console for UNIX mail servers** receives configuration from the **Dr.Web Agent** socket. If **Dr.Web Agent** is operating in the Standalone mode, the following warning is output to the **Dr.Web Console for UNIX mail servers**:
Unable to connect to Dr.Web Agent at local:%var_dir/ipc/.agent
- Auto – **Dr.Web Console for UNIX mail servers** operation mode is set according to the mode of **Dr.Web Agent**.

If there is a problem connecting to **Dr.Web Enterprise Server**, the following behaviours of **Dr.Web Console for UNIX mail servers** are possible:

- If **Dr.Web Enterprise Server** is unavailable upon the initial connection or authorization process fails, **Dr.Web Agent** terminates. In this case, check the settings and try to restart **Dr.Web Agent** and **Dr.Web Console for UNIX mail servers**.
- If connection to **Dr.Web Enterprise Server** was established earlier, but now the server is



temporary unavailable (for example, in the event of connection problems), **Dr.Web Agent** uses backup copies of configuration files that were previously received from the server. These files are encrypted and must not be edit by users. Edited files become invalid.

When **Dr.Web Console for UNIX mail servers** enters Enterprise mode, **(CPM)** (Central Protection Mode) label displays on the top navigation menu of **Dr.Web Console for UNIX mail servers**.



Figure 60. Dr.Web Console for UNIX mail servers operation mode

Configuring User Permissions

When **Dr.Web Agent** is running in the Enterprise mode, **Dr.Web Control Center** administrator can partially or completely block user permission to configure **Dr.Web** components installed on the workstation.

To set permissions of a workstation user:

- Enter **Dr.Web Control Center**. Note that the administrator must have sufficient privileges to adjust settings of **Dr.Web** anti-virus software.
- On the main menu, select **Network**, then click the workstation name in the hierarchical list. On the open control menu (left pane), select **Permissions**. This opens the permission configuration window.

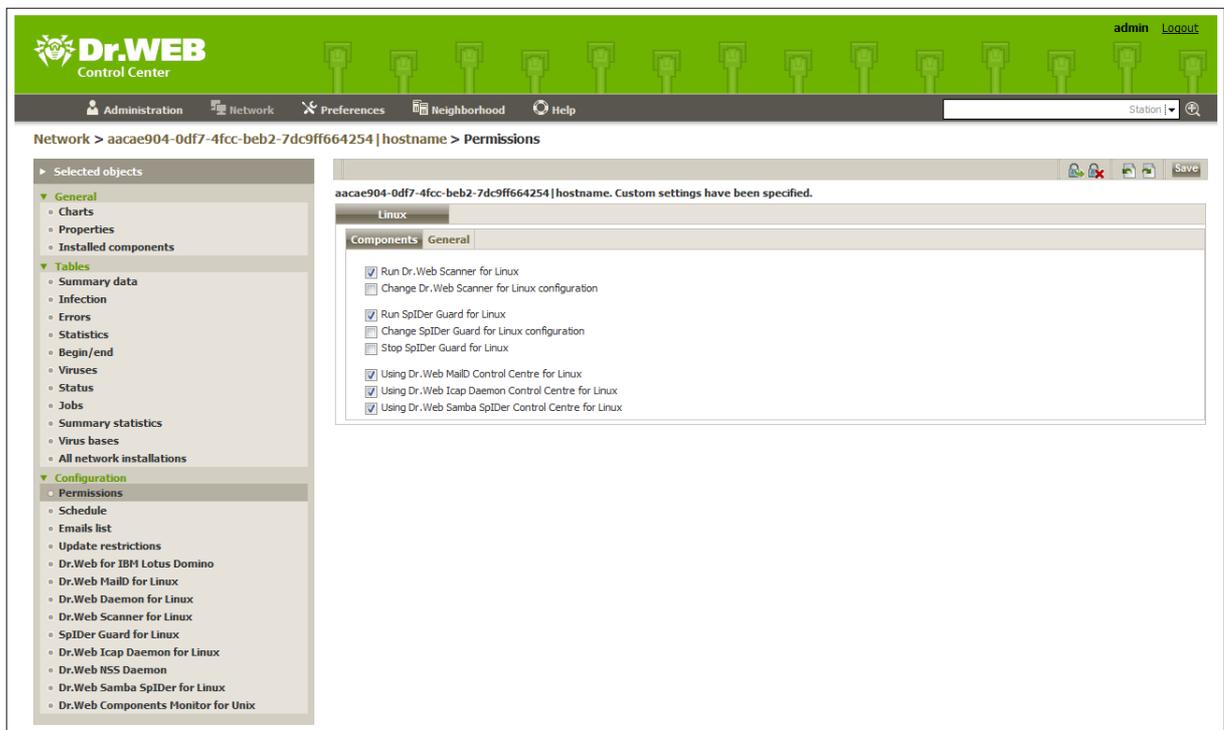


Figure 61. User permissions configuration

- In the **Components** section, select components to be available for the workstation user to change. For example, to allow the workstation user to adjust **Dr.Web for UNIX mail servers** configuration, select the **Using Dr.Web MailD Control Centre for Linux** checkbox and click



Save.

- To disable the workstation user to adjust **Dr.Web for UNIX mail servers** configuration, clear the **Using Dr.Web MailD Control Centre for Linux** checkbox and click **Save**. In this mode, **Console** displays the corresponding warning and **Apply and Save Settings, Preview** and **Save** buttons become unavailable.

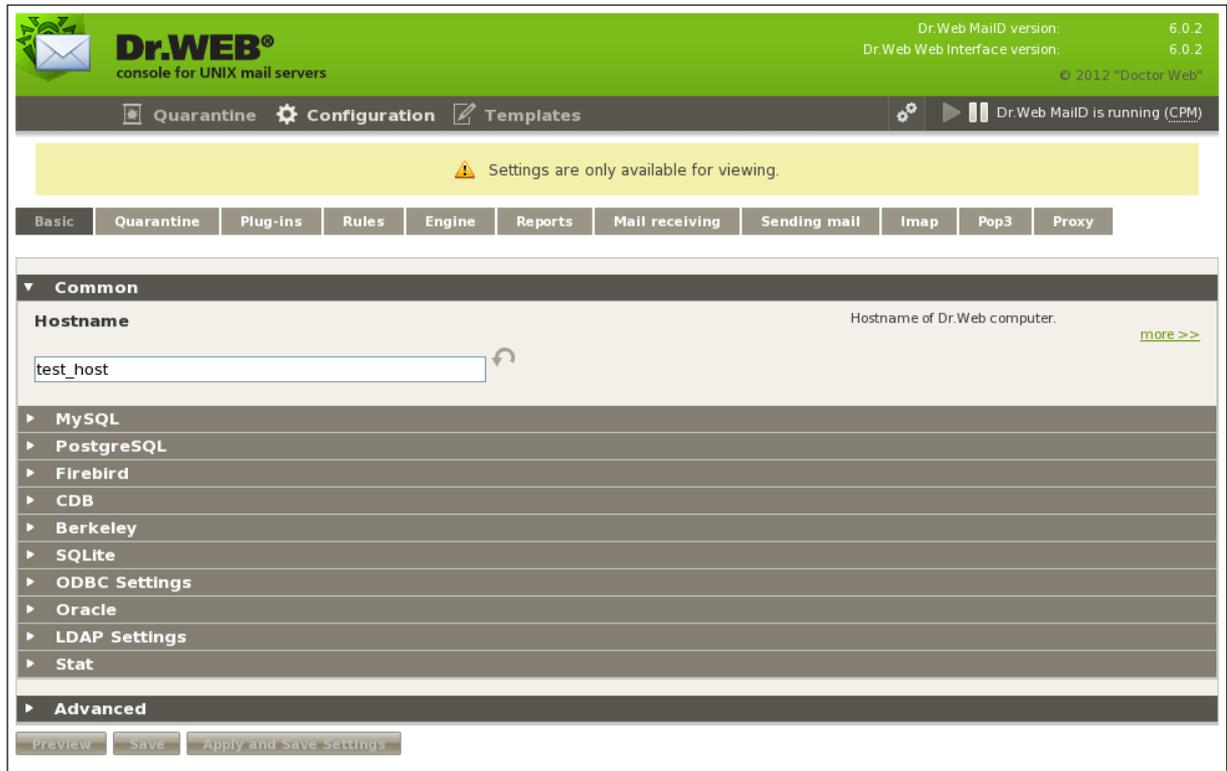


Figure 62. Read-only user permissions

Configuring Workstation

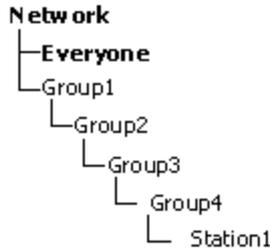
When a new workstation is created, its configuration settings are inherited from a group it belongs to. That group is called the *primary group*. If the settings of the primary group are modified, these changes are inherited by all workstations included into the group, unless the workstation configuration is customized. When creating a workstation, you can specify what group is to be treated as primary. By default, the primary group is the **Everyone** group.

Inheritance in nested groups depends on the group hierarchy. If for a station no custom settings are specified, it inherits configuration from its parent group, and this process repeats recursively. Therefore, search for the group configuration is performed upwards through the hierarchical tree of nested groups, starting from the primary group of the station and further until the root group is reached. If no custom settings are found, the workstation inherits configuration of the **Everyone** group.



Example:

The structure of a hierarchical list is as follows:



Group4 is the primary group for Station1. To determine the settings to be inherited by Station1, the search is performed in the following order: Station1 → Group4 → Group3 → Group2 → Group1 → Everyone.

You can edit configuration inherited from the primary group in two ways:

- Using **Dr.Web Control Center** interface. To edit configuration, select **Network** on the main menu, then click the workstation name in the hierarchical list. On the control menu (on the left pane), select the component you want to configure. You need the **corresponding permissions** to perform this operation. The configuration process is similar to the one via **Console**. When necessary changes are made, click **Save** to save them.

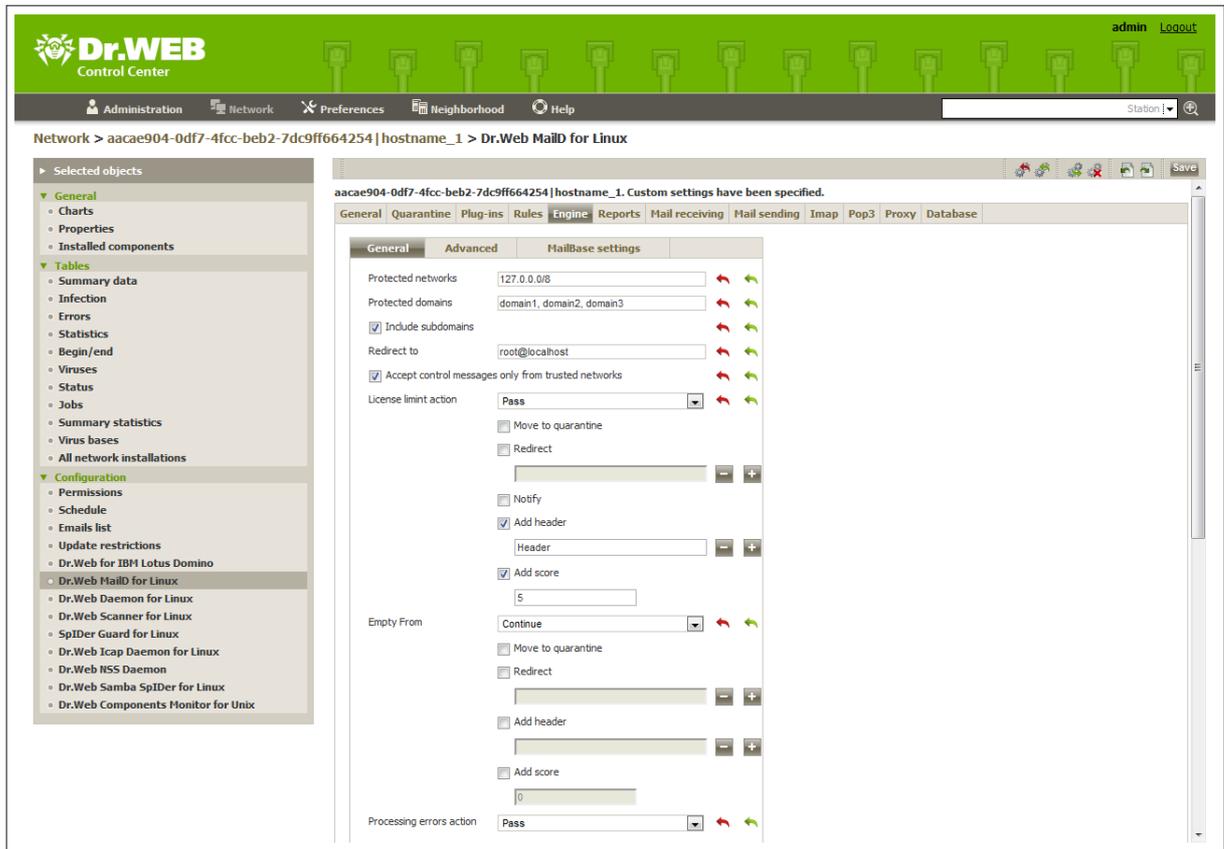


Figure 63. Configuration of Dr.Web MailD for Linux via Dr.Web Control Center interface

If appropriate permissions are set, parameters can be reconfigured via **Dr.Web Console for UNIX mail servers**. The configuration process is similar to the one in the **Standalone mode**. If the workstation user has insufficient privileges for that, settings are open in read-only mode.



Types of Administrator Accounts

There are four types of administrator accounts:

- *Administrators with full rights* have exclusive rights for management of **Dr.Web Enterprise Server** and **Anti-virus network**. They can view and edit the **Anti-virus network** configuration and create new administrator accounts. An administrator with full rights can configure the anti-virus software installed on the workstation, limit and disable user intervention into anti-virus software administration.

An administrator with full rights can view and edit the list of current administrator accounts.

- *Administrators with read-only rights* can only view **Anti-virus network** settings and its separate elements, but cannot modify them.
- *Group Administrators with full rights* have access to all system groups and those custom groups which they are allowed to manage (including nested groups). *Group Administrator* accounts can be created for custom groups only (see Administrator manual for **Dr.Web® Enterprise Security Suite**). In the hierarchical tree, only those groups are displayed for *group administrators* which they are allowed to access.

The list of current administrator accounts is not available for *Group Administrators*.

- *Group Administrators* with read-only rights can be granted full rights to adjust the available groups or read-only rights.
- *Default administrators* with full rights created automatically during **Dr.Web Enterprise Server** (the **admin** account).

Thus, *Administrators with full rights* can:

- Add new and delete already existing administrator accounts.
- Adjust settings for all administrators of **Anti-virus network**.

Group administrators and *administrators with read-only rights* can:

- Adjust some of their account settings.



Contacts

Dr.Web for UNIX mail servers solution is constantly improved. You can find news and the latest information on available updates on the website at:

<http://www.drweb.com/>

Sales department:

<http://buy.drweb.com/>

Technical support:

<http://support.drweb.com/>

Please include the following information in the problem report:

- full name and version of your operating system;
- versions of **Dr.Web for UNIX mail servers** modules;
- configuration files of all modules;
- log files of all modules.

